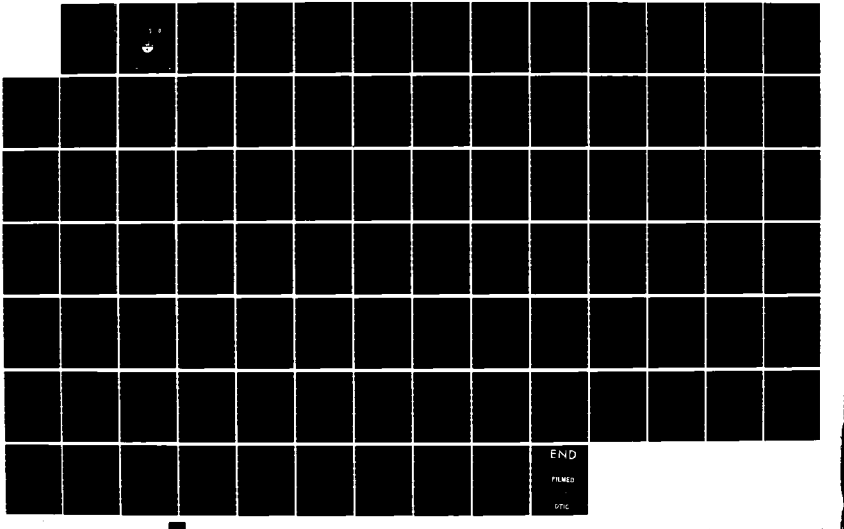
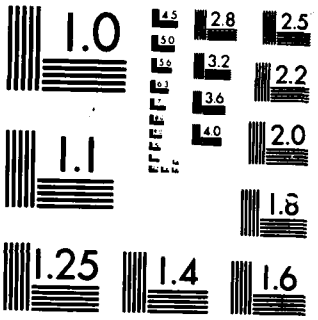
AD-A165 443 PROJECT STEAMER V MATHEMATICAL SIMULATION OF STEAMER PROPULSION PLANT(U) BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA B ROBERTS ET AL. AUG 81 NPRDC-TN-81-27 1/1

UNCLASSIFIED N00123-81-D-0794 F/G 5/9 NL





NPRDC TN 81-27

AUGUST 1981

PROJECT STEAMER: V. MATHEMATICAL SIMULATION OF STEAMER PROPULSION PLANT

NAVY PERSONNEL RESEARCH
AND
DEVELOPMENT CENTER
San Diego, California 92152

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NPRDC Technical Note 81-27 August 1981

PROJECT STEAMER: V. MATHEMATICAL SIMULATION OF STEAMER PROPULSION PLANT

Bruce Roberts
Kenneth Forbus

Bolt Beranek and Newman, Inc.
Cambridge, MA 02138

Reviewed by
John D. Ford, Jr.

Released by
James F. Kelly, Jr.
Commanding Officer

Navy Personnel Research and Development Center
San Diego, California 92152

FOREWORD

This research and development was conducted under contract N00123-81-D0794 with Bolt Beranek and Newman, Inc. in support of Navy Decision Coordinating Paper, Z1177-PN (Advanced Computer-Aided Instruction), subproject Z1177-PN.03 (STEAMER: Advanced Computer-Based Training for Propulsion and Problem Solving). It was sponsored by the Chief of Naval Operations (OP-01). The main objective of the STEAMER effort is to develop and evaluate advanced knowledge-based techniques for use in low-cost portable training systems. The project is focused on propulsion engineering as a domain in which to investigate these computer-based training techniques.

This report, the fifth in a series on the STEAMER project, describes the conversion of the 19E22 mathematical model into an interactive computer environment to serve as a base for subsequent efforts to produce intelligent personal training aids. Previous reports described an initial framework for developing techniques for automatically generating explanations of how to operate complex physical devices, provided a user's manual for the STEAMER interactive graphics package, described a method for generating explanations using qualitative simulation, and described CONLAN, a constraint-based programming language well suited for describing and analyzing complex devices (NPRDC TNs 81-21, 81-22, 81-25, and 81-26 respectively). Intended users of this report are system maintainers and other research personnel.

Appreciation is expressed to CAPT Neil Ammerman and his staff at the Surface Warfare Officers School in Newport, Rhode Island for several beneficial discussions about the nature of the training problem being addressed in this R&D effort, especially to LCDR Dan Bowler for providing information on how to operate and use the 19E22 simulator.

The contracting officer's technical representative was Dr. James D. Hollan.

JAMES F. KELLY, JR.
Commanding Officer

JAMES J. REGAN
Technical Director

SUMMARY

Problem

The main objective of the STEAMER effort is to develop and evaluate advanced knowledge-based techniques for use in low-cost portable training systems. The project is focused on propulsion engineering as a domain in which to investigate these computer-based training techniques. An important component of the STEAMER system is a detailed mathematical simulation of a propulsion plant.

Objective

This report describes the engine room portion of the STEAMER propulsion plant mathematical model and methods for interactively using it.

Approach

The STEAMER mathematical simulation is based on the 19E22 training device and has been implemented in Maclisp as a preparatory step in the development of a knowledge-based simulation of steam propulsion plant operations. Maclisp is an interactive language that provides facilities for observing and modifying running programs and for interfacing other modules to them.

Results

The math model is described, including methods for starting and stopping it at consistent points, initializing it, running only parts of it at a time, and connecting other programmed modules to it (e.g., modules that print out trends in the values of specific variables). Facilities to smooth interaction with the system are also described, such as an on-line data base of variables containing information about their units, range, the models in which they occur, and their role in the simulation (e.g., as internal constants or as values sent to the real-time interface).

CONTENTS

INTRODUCTION

Problem

The main objective of the STEAMER effort is to develop and evaluate advanced knowledge-based techniques for use in low-cost portable training systems. The project is focused on propulsion engineering as a domain in which to investigate these computer-based training techniques. An important base component of STEAMER is a detailed mathematical simulation of the propulsion plant operations. Such a mathematical simulation has already been developed by the Navy as part of the 19E22 training device.[1] For use in the 19E22, the simulation is implemented as a program that runs on a Harris computer to drive a mockup of the fire room, engine room, and auxiliary spaces of a FF1052/1078 class ship. It contains approximately 10,000 lines of FORTRAN code, not counting the real-time interface, operator console interface, or the operating system for the Harris computer.

Purpose

This report describes the engine room portion of the STEAMER propulsion plant mathematical model and methods for interactively using it.

APPROACH

In order to use techniques developed for performing intelligent computer-aided instruction (e.g., the abilities to run processes in different environments, for one process to examine and manipulate another process, or to maintain large evolving data structures), it is important to implement the mathematical model in a symbolic processing

---

[1]The 19E22 mathematical model has been published by the Naval Training Equipment Center as Technical Report NAVTRAEQUIPCEN 75-C-0115-38A.

language. The language MACLISP, an interactive language well suited for developing knowledge-based systems, has numerical capabilities comparable in power and efficiency to FORTRAN. MACLISP is widely used in the artificial intelligence community and has been used in many projects to increase computer intelligence and enlarge our understanding of human cognition. The future of MACLISP is guaranteed by the well-planned integration with its successors: NIL on the Digital Equipment Corporation VAX family, and Lisp Machine LISP on the MIT Lisp Machine, a prototype of the powerful personal computers to come.[2] For these reasons, the STEAMER mathematical simulation is based on the 19E22 training device.

RESULTS

Simulation Subsystems

The overall steam propulsion plant simulation comprises three separate subsystems: engine room, fire room, and auxiliary spaces. Each subsystem is designed to be run either independently or as part of an integrated simulation. We have completely implemented the engine room subsystem in MACLISP. The remainder of this section describes how to run the engine room models and interact with them in MACLISP. The basic style of running the models should not change appreciably as the remaining subsystems are added to it.

The engine room consists of 20 models that are executed repeatedly in a specified fixed order. Each model maintains a large number of state variables, both numerical (e.g., temperature, pressure, flow rate, and enthalpy) and logical (e.g., whether a pump is on or off, a valve open or closed). When a model is run, it updates some state variables to reflect the consequences of changes made by other models in the overall state of the steam plant and computes changes in variables due simply to the passage of time. The

[2]For a good introduction to MACLISP, see P. H. Winston and B. K. P. Horn, LISP, Addison-Wesley, 1980.

execution of a model is assumed to occur at a fixed interval, which is the value of the variable DELTAT.

Each variable is a MACLISP atom with a mnemonic name. Appendix A contains a list of the approximately 750 variables in the current system. A variable is evaluated to get its current value.[3]

The algorithm for each model in the simulation is implemented as a MACLISP function. Evaluating the function once recomputes the state variables of the model. Table 1 lists the models included in the engine room subsystem of the simulation.

Running the Simulation

The simulation described herein is currently available on System G at BBN in the file <STEAMER>ENGINEROOM.EXE. Running this file will recreate a MACLISP environment with all the engine room models loaded and initialized.

Alternatively, the simulation can be loaded into an existing MACLISP of one's own. Type:

```
(LOAD '|<STEAMER>LISP.INI|)
(ENGINEROOM)
```

to load the files containing the simulator and the variable data base.

The simulation consists of evaluating each of the models in the following sequence:

Electric
AlarmBoard (skipped)
ControlAir
150psiSteam
FireMain
AuxExhaust
DPumps
Distiller
SalinityPanel (skipped)
LOPump
LOPres
LOTemp

[3]A few arrays are also used to hold intermediate results. In MACLISP, arrays are accessed by using the array name as one would a function name in an expression with indices as arguments; for example, (LOPumpFlow 2) returns the second entry in the array LOPumpFlow.

Table 1

Models Included in Simulation Subsystem

Model	Function
150psiSteam	This model computes the availability and distribution of 150 psi steam in the engine room.
AirEjector	This model describes the main air ejector and gland exhaust condenser. It determines the alignment of both stages of the two air ejectors and the resulting vacuum produced in the main condenser.
AuxExhaust	This model computes values for the auxiliary exhaust in the engine room; primarily, for use by the main condenser unloading valve.
ControlAir	This model computes the supply of control air to the valves in the engine room.
Distiller	This model describes the distiller and the temperatures, pressures, and volumes of feedwater as it is used first as the coolant and then heated (by 3psig desuperheated steam) to 170 degrees Fahrenheit and flashed. 150 psi auxiliary steam is used with the two-stage air ejectors to produce the distiller vacuum.
DPumps	This model computes the status of the pumps associated with the distiller: the feed pump, the discharge pump, and the heater drain pump.
Electric	This model computes the distribution of power to motors in the engine room. It computes the total load on the electrical system as a function of circuit breaker positions and individual electrical loads.
Firemain	This model computes pressure in the firemain, the status of the fire pumps, and distribution to the main condenser and distiller.
JackingMotor	This model describes the operational status of the jacking motor attached to the main reduction gear of the turbine.
LOPres	This model includes operation of the lube oil strainer and computes pressures for the gages in the main lube oil system. Fluid flow is calculated to determine the main sump level.
LOPump	This model controls the motors on the two lube oil pumps, each with high and low speed, and manual or pressure controlled, direct or standby/emergency operation.
LOTemp	This model computes the temperature of the lube oil in the main engine as a function of flow and speed, and in the oil heater and oil cooler.
MainCondenser	This model computes the heat exchanged between the steam condensing from the turbine outlet and the sea water flowing through the condenser.
MainSteam	This model represents the alignment of the main steam system in the engine room during Warmup, Operating, and Secure modes. It also computes the steam demand and the main turbine header temperature and pressure.
MainTurbine	This model computes steam pressures and temperatures in the high and low pressure turbines. It determines the speed of the reduction gears, propeller shaft, propeller, and, using a model of the ship's dynamics, the overall speed of the ship.
MCCircPump	This model represents the electrical control of the main sea water circulating pump.
MCCondPumps	This model computes flow from the main condenser produced by the condensate pumps as a function of suction head, system load pressure and valve status. It also maintains the status of the condensate pumps.
MCRecirc	This model represents the automatic (thermal) or manual (by-pass) recirculating flow path for the condensate when required by a high condensate temperature.
MCSeaCirc	This model computes the sea water flow and pressures generated by the scoop and/or pump. They are used in the main condensate system.
MTGlandSeal	This model computes the steam availability and pressure (normally 2 psig) in the gland seals of the high pressure and low pressure turbines.

Note. The abbreviations for the engine room models were chosen to allow one to refer easily to the individual models while running the simulation.

MainSteam
JackingMotor
MTGlandSeal
MainTurbine
MainCondenser
AirEjector
MCRecirc
MCCondPumps
MCCircPump
MCSea Circ

Once started, MACLISP prints the prompt character ">." Any expression typed by the user in response to the prompt is immediately evalaluated by MACLISP. All evaluated expressions return a value, which is printed out by MACLISP before the next prompt.[4] Running the simulator consists of evaluating the proper sequence of model functions. The simplest way to run the simulation is with the function RUN, which has three forms:

1. (RUN) All of the models are executed repeatedly.

2. (RUN expression)
   Like (RUN) except:

   If expression is a number, the models are executed that many times; then the simulation halts.

   If expression is any other s-expression, the models are executed until (EVAL expression) is non-NIL; then the simulation halts.

3. (RUN expression models)
   Like (RUN expression), but only runs the models in models, a list of model names, or a single model name. Any collection of names presented in section 3.1 is recognized.

For example, to execute the AirEjector model 10 times, you would type (RUN 10 'AIREJECTOR); to run all the models until the shaft speed reaches 100 RPM, type (RUN '(> ShaftRotation 100.0)).[5]

---

[4]Some of the top level functions we have provided return a null character that does not show up when printed on the terminal.

[5]RUN is a Maclisp EXPR; that is, the arguments to RUN are evaluated before the function is executed. Thus, literal arguments other than numbers must be preceded by the MACLISP quote character "'".

Although RUN makes operating and monitoring the simulation easy, other means of controlling the simulation are provided to facilitate some common operations. Each of the models can be individually initialized, be turned on or off, and have its status printed. The mechanism for specifying these actions is designed to allow a user to think in terms of sending commands to the models telling them what should be done. Rather than having to remember all the functions written to carry out each action for each different model, one need only know the names of the models and the keywords to specify the action.

Models are conceived as entities having a limited number of capabilities to perform certain actions. The most commonly exercised capability is the ability to run the code that simulates the behavior of a portion of the steam plant. RUN works by successively sending the command "RUN" to each of the models in the engine room. Other commands are sent as follows:

(COMMAND command)

The command is sent to all the models in the engine room. The following are recognized currently:

LOAD	Load the file containing the interpreted version of the code for the model.
FASLOAD	Load the file containing the compiled version of the code for the model.[6] A compiled function runs faster, but cannot be inspected as it is running. The Engine Room starts out running only compiled functions.
INITIALIZE	Define and initialize the variables and arrays used in the model. This command can be used to revert the model to one of four known states, as defined by the value of the variable INITIALCONDITION (see pg. 9).
RUN	Execute the simulation function for the model once.
ON	Turn the model on by setting its variables to a normal operating condition.
OFF	Turn the model off by setting the variables to a normal secure condition.

[6]Files of compiled MACLISP code are called "fasl" files and can be recognized by having this as their file name extension.

STATUS — Print out the values of some selected variables that summarize the state of the model.

NAME — Print a long readable name for the model instead of the abbreviated name used to identify the model to COMMAND and RUN.

DESCRIBE — Print out a short description of the model's purpose and operation.

(COMMAND command models)
Like (COMMAND command), except send the command command to the model or list of models. (COMMAND command ?) prints the models that have the ability to accept the particular command. If a model is sent a command that it is unable to obey, nothing happens.

Because sending commands to models is such a common need, C, a version of COMMAND that does not evaluate its arguments, has been provided.[7] For example, (C ON DISTILLER) runs the function that turns on the Distiller. (C STATUS MAINTURBINE) runs the Status function for the *MainTurbine* model.

Controlling the Simulation

Some variables in the system are not part of any model but, rather, are used to select among alternate ways of reporting errors and initializing variables. Other variables interact with RUN to control details of its behavior.

The simulator recognizes two special conditions: freeze and malop. A freeze occurs when the simulator detects a value outside the range that can be simulated properly; for example, a casualty in which the lube oil unloading valve fails closed continues uncorrected. The simulator is not designed to continue after a freeze and should be reset to a known state.

A malop occurs when the simulator detects improper operation of the steam plant; for example, if a pump is turned on and its outlet valve is open but the valve controlling its gland sealing is closed. A large number of specially chosen malops are recognized in the simulator.

---

[7]Like the RUN function, the arguments to COMMAND are first evaluated. Consequently, literal arguments must be preceded by the quote character "'".

When a freeze or a malop is encountered in the simulation, the value of the FREEZE or MALOP variable, respectively, controls the action taken as follows:

1. FREEZE (normally set to BREAK)

   NIL — The freeze is ignored and the computation continues.

   WARN — A message is printed, but the computation continues.

   BREAK — A message is printed and a sequence break occurs. Proceeding from the break (by typing <altmode>P<space>) causes the simulation to continue.

2. MALOP (normally set to WARN)

   NIL — The malop is ignored and the computation continues.

   WARN — A message is printed, but the computation continues.

   BREAK — A message is printed and a sequence break occurs. Proceeding from the break (by typing <altmode>P<space>) causes the simulation to continue.

Each time through the simulation, the function RUN does the following:

1. Increments the variable TIME by the value of DELTAT. DELTAT is 1.0, so TIME records the total number of iterations through the simulation.

2. Evaluates all forms in the list STATUS. For example, to monitor a variable's trend, an expression is put in the STATUS list to print out its value each iteration. Expressions in STATUS can do arbitrary calculations. RUN simply evaluates them at the end of each pass through the simulation. One sometimes useful expression to put on the STATUS list is (WAIT n). WAIT pauses until n seconds after the previous call to WAIT have elapsed. Adjusting n controls the speed of the simulation in real time.

3. Checks the variable STOP?. If STOP? is NIL, RUN steps through the simulation again. If STOP? is T, RUN causes a sequence break and returns control to the terminal so that you can inspect or modify the state of the simulation. Continuing from the breakpoint by typing <altmode>P<space> resumes the simulation. If STOP? is still non-NIL, of course, the simulation will again stop after one more pass through all the models.

Control-N can be typed anytime to start and stop the simulation. The Control-N interrupt has been set to toggle the value of STOP? between T and NIL. If Control-N is typed while at a breakpoint caused by the RUN function, in addition to switching the value of STOP?, the simulation continues immediately from the breakpoint.

One can always interrupt processing in Maclisp by typing Control-B. This causes an immediate sequence break the moment the character is typed. Continuing from a breakpoint resumes processing from the point it was interrupted. Setting STOP? lets one break at a point in the simulation where all models have been updated.

STATUS forms are evaluated before the function RUN checks STOP?, so a STATUS form can set STOP? to NIL and thereby halt the simulation. Any remaining STATUS forms will be evaluated first. For example, to print out the hotwell temperature and stop the simulation if it goes over 75 degrees, do the following:

```
(SETQ STATUS (LIST '   (PRINTVARIABLE 'HOTWELLTEMP))
                       '(COND     ((> HOTWELLTEMP 75.0)
                                   (SETQ STOP? T)))))
```

The models can be inintialized to one of several stats, as defined by the value of the of variable INITIALCONDITION (see pg. 6). Possible values include:

1. COLDIRON Dockside, with shore steam and shore power.

2. AUXSTEAM Dockside, with one boiler and two ship service turbogenerators, but no engine operating.

3. 15KNOTS Steaming at 15 knots, with one boiler under automatic control, two ship service turbogenerators in parallel service, and the diesel generator lined up for automatic start.

4. 25KNOTS Steam at 25 knots, with two boilers; otherwise like 15KNOTS.

If QUIET? is NIL, and there are no STATUS forms to be evaluated, the function RUN prints out "tick" each time through the simulation. Setting QUIET? to T inhibits this feature of RUN.

Monitoring the Simulation

The simulation is monitored by keeping track of a collection of variables that represent some portion of the overall state of the steam plant. While this can be done simply by periodically printing on the terminal the name of a variable and its value, a better method is available. The display screen of your terminal is divided into regions, or windows, by typing (PANEL). Normal terminal dialogue appears in a window at the bottom of the screen called the Lisp-Window. The top of the screen is another window called the BannerBoard, which is reserved for displaying variables, their name, and value. The middle of the screen is a third window called the SwitchBoard, which can be used to display variables that have only two possible values (ON or OFF, for example). Rather than printing out the value of the variable like the BannerBoard, the SwitchBoard depicts values by changing the background color of the label. The SwitchBoard itself is grey; values are shown as either white or black.

The following functions provide the means for creating and updating panels to display particular sets of variables:

1. (CREATE-PANEL banners switches) creates a panel. It returns a list that is the argument for the following two functions: RESET-PANEL and SHOW-PANEL. Banners and Switches are lists of variables, either atoms or pairs, whose first element is an any Lisp expression and second element is a label for the expression's value shown on the screen. The expression or variable will be evaluated each time the panel is updated.

2. (RESET-PANEL panel) clears the screen and draws the static portion of the panel; that is, the labels.

3. (SHOW-PANEL panel) updates the panel by writing the new value of any changed variables in the BannerBoard, and changing the color of the changed variables in the SwitchBoard. Panel should be the one currently being displayed; that is, the one last initialized with RESET-PANEL. For example,

```
(CREATE-PANEL
    '(MCSWCondFlow ((-$ 13500.0 CSWCondFlow) . "Reserve C
    '(PumpButterflyValve PumpFlapperOpen))
```

creates a panel that shows on the BannerBoard the value of MCSWCondFlow[8] and the difference between that variable and the known capacity of the sea water circulating pump. The difference is labeled "Reserve Capacity." The SwitchBoard shows the values of PumpButterflyValve and PumpFlapperOpen. This example demonstrates the ability to define and monitor conditions that are not directly present in the math model as it stands, but which can be computed from existing variables.

While many panels can be defined, only one can be visible at a time. Using CREATE-Panel, many panels can be created to correspond to different collections of state variables. Any one panel can then be displayed using RESET-PANEL and subsequently updated using SHOW-PANEL. A common thing to do is to create a panel, assign it to a variable (say P1), and append to STATUS the form (SHOW-PANEL P1). Henceforth, the displayed variables will be updated each time through the simulation and the panel window will always contain the current state while the simulation is running.

Looking up Variables Used in the Simulation

The simulator variables have been given mnemonic names to facilitate remembering them and recalling their meaning. Appendix A is a list of variables currently used in the simulation. A data base of variables residing in the system stores the following useful information:

1. Fortran name identifying the variable in the math model documentation.
2. Initial condition, or constant value.
3. Units.

---

[8]MCSWCondFlow is the flow of sea water through the main condenser measured in GPM.

4. Normal range.

5. Models in which the variable occurs.

6. Type: Model constant, internal, exported, imported, or real-time interface variable.

7. Discrete or continuous.

The data base is also used to store a string representation of the name for more readable output[9] and to allow automatic completion of variable names during input.

The following functions can be used to help identify variables by looking for names that contain particular strings of characters.

1. (AproposVariable substring1 . . . substringN)
   Returns a list of variables that each contain all the substrings.

2. (AproposVariablePrint substring1 . . . substringN)
   Like AproposVariable, but prints out the names instead of returning them as a list.

It is often useful to find variables that share common properties; for example, all the variables in the AirEjector model used for the real-time interface. The following functions retrieve variables from the data base.

1. (FindVariable keywords models)
   returns a list of variables used in models that contain each of keywords in their model description. Model descriptions characterize each variable's use in a model with one of the following phrases, in which the vertical bar "|" denotes "or":

   - model constant
   - discrete | continuous from | to models | RTI
   - discrete | continuous internals

2. (FindVariablePrint Keywords models)
   Like WhichVariable, but prints out the names instead of returning them as a list.

---

[9]MACLISP converts each character to upper case during input unless it is individually quoted with "/" or occurs within an atom delimited by "|".

Information about individual variables can be obtained using the following functions:

1. (DV name) Describes the variable. It prints information from the variable data base.
2. (DFV name) Like DV, but accepts the variable's Fortran name instead.
3. (PV name) Prints the variable's current value.
4. (PFV name) Like PV, but accepts the variable's Fortran name instead.

Since names for variables were chosen to be descriptive of the value they represent, they tend to be very long. A mechanism for automatically supplying the remainder of a partially typed name has been provided to facilitate entering these lengthy names at the toplevel of Lisp. Whenever the character "%" is encountered by the MACLISP reader, it enters a special mode expecting a variable name to be typed. Within this mode, the following characters are treated specially:

1. SPACE Completes the entry if a variable name is unambiguously specified by the characters entered. Otherwise, the bell is rung and the reader waits for further characters to disambiguate the partial name.
2. RETURN This is treated just like a Space.
3. RUBOUT Deletes the last character typed. Deleting the "%" character exists from variable reading mode.
4. ? Lists the possible completions.
5. Control-F Completes the name to the first ambiguous character.
6. Control-W Deletes all characters entered so far, but stays in variable reading mode.
7. Control-U Exits variable reading mode.
8. Control-R Retypes the characters in the name.

Timing Measurement for the Simulation

The Maclisp simulation has been run in on a Digital Equipment Corporation KL-10. One hundred iterations of RUN took approximately 1.5 seconds, and included one garbage collection to reclaim FLONUM space. The simulation was running at its maximum rate.

No delays were introduced to provide "real-time" behavior. Although even this speed could be further improved,[10] there is already sufficient headroom for considerable additional computation. The speed of the simulation need not be compromised to build a simulation that would be easy to modify and connect to other components of the overall system.

[10]By storing all variables in arrays rather than as individual atoms.

REFERENCES

Forbus, K. D. Project STEAMER: IV. A primer on CONLAN--A constraint-based language for describing the operation of complex physical devices (NPRDC Tech. Note 81-26). San Diego: Navy Personnel Research and Development Center, August 1981.

Forbus, K. D., & Stevens, A. Project STEAMER: III. Using qualitative simulation to generate explanations of how to operate complex physical devices (NPRDC Tech. Note 81-25). San Diego: Navy Personnel Research and Development Center, August 1981.

Stead, L. Project STEAMER: II. User's manual for the STEAMER interacative graphics package (NPRDC Tech. Note 81-22). San Diego: Navy Personnel Research and Development Center, August 1981.

Stevens, A., & Steinberg, C. Project STEAMER: I. Taxonomy for generating explanations of how to operate complex physical devices (NPRDC Tech. Note 81-21). San Diego: Navy Personnel Research and Development Center, August 1981.

APPENDIX A

LIST OF VARIABLES

LIST OF VARIABLES

12001bAuxSteam
12001bAuxSteamGage
1501bAuxSteamGage
1501bAuxSteamGage1
1501bAuxSteamGage2
1501bSteamAvailable
1501bSteamEnable
1501bSteamEnable?
1501bSteamEnthalpy
1501bSteamPres
150psiSteamEnthalpy
150psiSteamMinValue
3psigSteamControlValve
3to15psigAirPilotOutputPres
AccelMode1
AccelMode2
AccelMode3
AdvanceCoeffFunction
AftDevices
AftJournalBearingTemp
AftLamps
AheadEnable
AheadGuarding
AheadGuardingWarmupBypass
AheadSteamChestPres
AheadSteamChestPresTC
AirEjectorAuralCue
AirEjectorBoilingTemp
AirPilotInput
AmbientHeatLossCoeff
AsternEnable
AsternGuarding
AsternGuardingWarmupBypass
AsternSteamFlow
AsternSteamFlowFunction
AsternThrottle
AttachedLOPumpFlow
AttachedPumpDischargeF/PDCoeff
AttachedPumpDischargePres
AttachedPumpSuctionF/PDCoeff
AttachedPumpSuctionPres
AuralCueCircPumpHi/Lo
AuralCueCircPumpOn/Off
AuralCueEnableVentFans
AuxExhaustHeaderStopValve
AuxExhaustPres
AuxExhaustSteamValve
AuxExhaustSteamValve1
AuxExhaustSupplyPres
AuxExhaustToDistilPlant
AuxExhaustToDPCAPres
AuxExhaustUnloadingPres
AuxExhaustUnloadingWeightFlow
AuxMachSWCoolingOn
AuxSteamToDAEPres
AuxSteamToOrificePres
AuxSWCoolingHeaderOn
BearingsF/PDCoeff
BearingTC
BearingTempFactor
BearingTempRise
BoilerlASteamDrumLevelIndicator
BoilerlBSteamDrumLevelIndicator
BrineEductorSound
BrineOverboardDischargePres
BrineOverboardValve
BypassValveCoeff
CasualtyAftJournalBearing
CasualtyAttachedPumpFail
CasualtyCircPumpMotorFail
CasualtyCloggedStrainer
CasualtyCondPumplA
CasualtyCondPumplB
CasualtyFwdJournalBearing
CasualtyFwdPinionBearing
CasualtyGlandSealLoss
CasualtyGlandSteamLoss
CasualtyInstructorRumble
CasualtyLOCoolerFailure
CasualtyLOPumpl
CasualtyLOPump2
CasualtyMainThrottleInoperative
CasualtyMainThrustBearing
CasualtyPresAmplitude
CasualtyRecircValveFailClosed
CasualtyReductionGearNoise
CasualtyScoopFlapperValveJamClosed
CasualtyScoopFlapperValveJamOpen
CasualtyStagelUnitlAEFail
CasualtyStagelUnit2AEFail
CasualtyStage2Unit1AEFail
CasualtyStage2Unit2AEFail
CasualtyTC
CasualtyUnloadingValveFailClosed
CasualtyUnloadingValveStuckOpen
CasualtyUnloadingValveStuckShut

CavitationPumpFlowLimit
Cell6DumpValveOverride
CentralDevices
CentralLamps
CircleWVentFansAS
CircleWVentFansASLamp
CircleWVentFansASPowerEnable
CircleWVentFansEC
CircleWVentFansECLamp
CircleWVentFansECPowerEnable
CircPumpHighSpeedIndicator
CircPumpHighStart
CircPumpLowSpeedIndicator
CircPumpLowStart
CircPumpMotorSpeed
CircPumpPowerOn
CircPumpStop
ColdStartFailureTC
CombinedTempInterceptConstant
CombinedTempSlopeConstant
CommonCondenserFill
CondensateFlow
CondensateFlowAECondenser
CondensateFlowAEEnthalpy
CondensateLevelGain
CondensateManometerBottomLevel
CondensateManometerLevel
CondensateOutSWHeaterTemp
CondensatePumpGlandSealLoss
CondensateRecircPathClosed
CondensateSystemLoadPres
CondensateSystemShutoffValve
CondensateSystemValve
CondensateToBilge
CondensateToBilgeValve
CondensateToCondensateSystem
CondensateToFireRoomFlow
CondensateToSteamDrainSystem
CondensateVolumetricFlow
CondenserAirPres
CondenserFillFlow
CondenserHeatTransferCoeff
CondenserLiquidTC
CondenserRecircValveHi
CondenserRecircValveLo
CondenserSteamPres
CondenserWaterBoxPres
CondPump1AAuralCue
CondPump1APowerOn
CondPump1BAuralCue
CondPump1BPowerOn
CondPumpNetPosSuctionHead
ControlAirGage
ControlAirGage2
ControlAirLinedUp
ControlAirOn
ControlPanelACCOn
CoolerCasualtyLeakConstant
CoolerCasualtyLostLubeFlow
CoolerConstant
CoolerF/PDCoeff
CoolerLeakF/PDCoeff
CoolingFlowIntercept
CoolingFlowThreshold
CoolingWaterPresHeader
CoolingWaterTemp
CounterLimit1
CounterLimit2
CounterLimit3
CrossConnect1
CrossConnect2
CrossConnect3
CrossConnect4
DischargeButterflyValvePos
DischargePres
DischargeTempCurveConstant
DischargeTempCurveSlopeConstant
DischargeTempInterceptConstant
DischargeValveOn
DistillateAvailable
DistillateLevelError
DistillateLevelFallGain
DistillateLevelGain
DistillateLevelInput
DistillateLevelInputGain
DistillateLevelOutput
DistillateLevelOutputGain
DistillateLevelRiseGain
DistillateManometerBottomLevel
DistillateManometerLevel
DistillatePumpDischargePres
DistillatePumpMotorOn
DistillatePumpOutletValve
DistillatePumpPowerOn
DistillatePumpSealValve
DistillatePumpSound
DistillatePumpStart
DistillatePumpStop
DistillateZeroPresLevel
DistillerBrineDischargeOn

DistillerBrineOverboardGage
DistillerPumpsEC
DistillerSteam
DistillerSteamGage
DownstreamValvePres
DPAuxExhaustDemand
DPAuxExhaustEnabled
DPresInput
DPresResponseError
DPresResponseGain
DPresResponseOutput
DTemp3
DTemp4
DTemp5
DTemp6
DTemp7
DTemp8
DTempInput
EBCSTM
EffectiveSeaWaterFlow
EK2TAU
EKR
ElectPumpDischargeF/PDCoeff
ElectPumpSuctionF/PDCoeff
EmergencyCondenserSupplyHeaderOn
EmergencyCoolingDieselGenOn
EmergencyWarmupConstant
EmergencyWarmUpF/PDropCoeff
EmergencyWarmUpFlow
EMKW
EMT1
EMT2
EMT3
EMT4
EMT5
EngineRoomSeaWaterTemp
EngineRoomSteam
EngineRoomSteam1
EstablishValueofP264
EstablishValueofP284
EstablishValueofV357
FeedPumpMotorOn
FeedPumpPowerOn
FeedPumpStart
FeedPumpStop
FeedwaterCoolerPres
FeedwaterInAfterCondenserTemp
FeedwaterInletTemp
FeedwaterInterstageTemp
FeedwaterOutAfterCondenserTemp
FeedwaterOutRefTemp
FeedwaterPresToStage1
FeedwaterPumpOutletPres
FeedwaterStrainerFeedSuction
FillFlowRateConstant
FilteredHeatExchangerValue
FinalDistillateTemp
FireMainHoseAdaptorValve
FireMainPres
FireMainStopValve1
FireMainStopValve2
FireMainStopValve3
FireMainToAuxSWCooling
FireMainToDistilBrineEductors
FirePump1Header
FirePump2Header
FirePump3AS
FirePump3ASLamp
FirePump3ASPowerEnable
FirePump3Control
FirePump3Control1
FirePump3Control2
FirePump3DischargeGage
FirePump3EC
FirePump3ECLamp
FirePump3ECPowerEnable
FirePump3Header
FirePump3MotorOn
FirePump3PowerOn
FirePump3SuctionGage
FirePump3SuctionValve
FirePump4Header
FirePumpOn
FireRoomCondensatePres
FireRoomDistillateValve1
FireRoomDistillateValve2
FirstPassFlag
ForwardSteamFlow
ForwardThrottle
FwdJournalBearingTemp
FwdPinionBearingTemp
FWDPumpOutletValve
FWDSeaChestValve
FWDTempControlValve
GlandExhaustEjector
GlandExhaustEjectorValve
GlandManifoldSteamPres
GlandSealFailureTC
GlandSealSteam
GlandSteamEnthalpy

GlandSteamFlow
GlandSteamLeakoffCoeff
GlandSteamPresTC
Heater/PurifierFlowPathEnabled
Heater/SteamHeaderPathEnabled
HeaterDrainRegJackSpindleOpen
HeaterLevelError
HeaterLevelFallGain
HeaterLevelGain
HeaterLevelInput
HeaterLevelOutput
HeaterLevelRiseGain
HeaterTempFlowFactor
HeaterTempRise
HeatExchangerFunction0
HeatExchangerFunction1
HeatExchangerFunction2
HeatExchangerFunction3
HiRecircHeatLossConstant
HotwellBottomPresHead
HotwellLevelGage
HotwellLiquidMass
HotwellLiquidVolume
HotwellTemp
HotwellWaterLevel
HPTurbineExhaustPres
HPTurbineExhaustTemp
IMPresTrans1
IMPresTrans2
IMPresTrans3
IMPresTrans4
IndependentModeAuxExhaustEnthalpy
IndependentModeAuxExhaustPres
IndicatorV595
InstructorEnableMainCondenserUnloading
InstructorSalinityKnob
JackHandleOperative
JackingGearForward
JackingGearReverse
JackingMotorStatus
JMBrake
JMEngage
JMFirstEngageFlag
JMLapsedTC
JMPowerOn
JMRelayFlag
JMRumbleCurveIntercept
JMRumbleCurveSlope
JMStartForward
JMStartReverse
JMStop/Reset
JMTC
LatchRelayFlag
LatchRelayFlag2
LightingEnable
LightingPowerEnable
LocalPumpHeader
LOContaminateDrainValve1
LOContaminateDrainValve2
LOContaminateDrainValve3
LOContaminateDrainValve4
LOCoolerInletPres
LOCoolerSWFlow
LOCoolerSWToMCRecircFlow
LOCoolerSWVent
LOCoolerTC
LOFlowInHPTurbine
LOFlowInLPTurbine
LOFlowInReductionGear
LOHeaterFlow
LOHeaterOn
LOHeaterSteamDrain
LOHeaterTC
LOHeaterTempDecayTC
LOHeaterValve1
LOHeaterValve2
LOHeaterValve3
LOHeaterValve4
LOHeaterValve5
LOLowPresAlarm
LOOutletTemp
LOPres
LOPump1DischargePres
LOPump1Flow
LOPump1Hi/Lo
LOPump1HighIndicator
LOPump1HighStart
LOPump1LowIndicator
LOPump1LowStart
LOPump1On
LOPump1Stop
LOPump1SuctionPres
LOPump2DischargePres
LOPump2Flow
LOPump2Hi/Lo
LOPump2HighIndicator
LOPump2HighStart
LOPump2LowIndicator
LOPump2LowStart
LOPump2On

LOPump2Stop
LOPump2SuctionPres
LOPumpsPowerOn
LOPurifierHeaterSteam
LOPurifierHeaterSteamGage
LOPurifierHNValveSteam
LOPurifierMotorON
LOPurifierPowerOn
LOPurifierStart
LOPurifierStop
LOPurifierValve1
LOPurifierValve2
LOPurifierValve3
LOPurifierValve4
LOPurifierValve5
LOPurifierValve6
LOPurifierValve7
LOPurifierValve8
LOPurifierValve9
LOPurifierWasteDrain
LOServiceMode
LOServicePumpFlowCommand
LOServicePumpFlowTC
LOServiceUnloading
LOServiceUnloadingValve
LOStorageTankValve1
LOStorageTankValve2
LOStorageTankValve3
LOStorageTankValve4
LOStorageTankValve5
LOSWCooler
LOSystemCapacity
LOTemp
LOTempCoeff
LOTurbineInletPresFlowTC
LOUnloadingValveFlowTC
LowSumpLevel
LPSentinelValveAuralCue
LPTurbineExhaustEnthalpy
LPTurbineExhaustPres
LPTurbineExhaustPresCondenser
LPTurbineExhaustTemp
LPTurbineInletPresAstern
MainCondenserCoolingValve
MainCondenserSteamEnthalpy
MainCondenserSteamFlow
MainCondenserUnloadingStopValve
MainCondPumpsStbyLoSpeed
MainHeaderAvailable
MainShaftNormalizedRPM
MainSumpOilTemp
MainThrottleAsternValveLift
MainThrottleForwardValveLift
MainThrustBearingTemp
Manual/AutoSelect
ManualBypassValveCoeff
ManualFlowBypassValveCoeff
ManualRecircBypassValve
MaximumPumpFlow
MaxRecircFlowSinglePump
MCCircPumpAS
MCCircPumpASLamp
MCCircPumpEC
MCCircPumpECLamp
MCFillStopValve
MCPumpASPowerEnable
MCPumpECPowerEnable
MCPumpsASPowerEnable
MCPumpsECPowerEnable
MCPumpsLPStbyHiSpeed
MCSWComplianceConstant
MCSWCondFlow
MCSWCondPres
MCSWPumpCoeff
MCSWResConstant
MCSWValveCoeff
MCSWValveFactor
MCUnloadingValveCommandedValveLift
MCUnloadingValveFlowPathEnabled
MCUnloadingValveLineupStatus
MCUnloadingValveManOverrideValveLift
MCUnloadingValveManualMode
MCUnloadingValveStatus
MCUVMMWeightFlow
MinValveSupplySteamPres
MinValveSupplySteamTemp
MSCasingDrain1
MSCasingDrain2
MSConstantDrain
MSCrossConnect
MSDecaySteamPres
MSDrainAlign
MSHeaderPres
MSSteamDemand
MSSteamEnable
MSSteamPres
MSSteamTemp
MSValveStroke
MSValveStrokeForward
MSWarmupDrain1

SecuredCondPres
SenseSupplyValve
SenseUnloadingValve
SenseUnloadingValve2
ServiceManualModeLift
ShaftFrictionCoeff1
ShaftFrictionCoeff2
ShaftFrictionCoeff3
ShaftLockingSpeedWindow
ShaftRotation
ShaftTorqueAstern
ShaftTorqueForward
ShipAdvanceCoeff
ShipResistance
ShipSpeed
Stage1AESteamValve
Stage1FailureTC
Stage1HPPres
Stage1HPTemp
Stage1ShellPres
Stage1ShellTemp
Stage1Unit1AELinedUp
Stage1Unit1AirSuctionValve
Stage1Unit2AELinedUp
Stage1Unit2AirSuctionValve
Stage2FailureTC
Stage2RefPres
Stage2ShellPres
Stage2ShellTemp
Stage2Unit1AELinedUp
Stage2Unit2AELinedUp
Standby/Emergency
StartCondPump1A
StartCondPump1B
StaticCondPresReading
StaticPumpDischargeReading
StaticPumpSuctionReading
SteamAverageEnthalpy
SteamChestContinuousDrain
SteamChestWarmupBypass1
SteamChestWarmupBypass2
SteamControlAheadGuarding
SteamDrainSystemValve
SteamEnthalpyValue
SteamFlowAstern
SteamFlowConstantReverse
SteamFlowModified
SteamFlowRatio
SteamFlowRatioForward
SteamInletValve
SteamInletValvesLinedUp
SteamPresKnob
SteamPresLevel
SteamToSWHeater
StemPositionUnloadingValve
StopCondPump1A
StopCondPump1B
StopValve1MS2A
StopValve1MS2B
StrainerCasualtyConstant
StrainerConstant
StrainerDrainValve1
StrainerDrainValve2
StrainerDrainValve3
StrainerF/PDCoeff
StrainerFlag
StrainerInletPres
StrainerJackingScrewsOnCap
StrainerLockingLugsOnFilter
StrainerLockingLugsOnFilter2
StrainerShiftingLever1
StrainerShiftingLever2
StrainerTrapDoorOpens
StrainerVentValve1
StrainerVentValve2
StrokeCrossbarMainThrottleForward
SuctionPres
SuctionValveOn
SupplyAirV595
SupplyBypassValve
SupplySteamPresFunction
SupplySteamTempFunction
SupplySteamTransientTemp
SupplyValveActuationAirPres
SupplyValveAirPres
SupplyValveCAPres
SupplyValveStrokeIndicator
SWCoolerOutletValveLift
SWHeaderPathEnabled
SWHeaterDrainPumpDischargePres
SWHeaterDrainPumpMotorOn
SWHeaterDrainPumpPowerOn
SWHeaterDrainPumpSealValve
SWHeaterDrainPumpSound
SWHeaterDrainPumpStart
SWHeaterDrainPumpStop
SWHeaterShellPres
SWHeaterShellTemp
SWOutletTemp
SWPathEnable

SWPres
SWPumpSuctionPres
SystemMode
TempDecayResponseGain
TempFlowTimeFactor
TempResponseError
TempResponseGain
TempResponseOutput
TempRiseResponseGain
TempTC
TempTC1
TempTC2
ThermostaticValveCoeff
TimeIncrement
TotalBearingF/PConstant
TotalCondPumpHead
TotalFlowHeatFunction
TotalHeadDFTBase
TotalLoadAft
TotalLoadCentral
TotalLOPumpFlow
TotalOilHeaderFlow
TotalShaftTorque
TotalSteamFlowInput
TrainerFreeze
TrialCondenserFlow1
TrialCondenserFlow2
TrialCondenserFlow3
TrialCondenserPres1
TrialCondenserPres2
TrialCondenserPres3
TSFlowRes1
TSFlowRes2
TSFlowRes3
TSPres1
TSPres2
TSPres3
TurbineDeltaEnthalpyExit
TurbineDeltaPresExit
TurbineExhaustEnthalpy
TurbineExitEnthalpy
TurbineHeaderPres
TurbineHeaderTemp
TurbineInletSteamPres
TurbineInletSteamTemp
UnfilteredHotwellTemp
UnfilteredSWOutletTemp
UnfilteredVaporTemp
Unit1AEInoperative
Unit1AEOperating
Unit1AirSuctionValve1
Unit1AirSuctionValve2
Unit2AEInoperative
Unit2AirSuctionValve1
Unit2AirSuctionValve2
UnloadingFlowDPEnabled
UnloadingValveActuationAirPres
UnloadingValveAirPres
UnloadingValveCAPres
UnloadingValveCoeff
UnloadingValveF/PDropCoeff
UnloadingValveFlowCoeff
UnloadingValveMOCoeff
UnloadingValveStrokeIndicator
UnloadingValveToSumpFlow
VacuumThreshold
VacuumVaporPres
ValveAlignedForRecirc
ValveCAPresMax
ValveCAPresMin
WarmupBypass1MS2A
WarmupBypass1MS2B
WarmupTC
WaterMassDerivative

APPENDIX B

PROGRAM LISTING

PROGRAM LISTING

B.1 150psiSteam

```
(defun 150psiSteamRun ()
    (SETQ 150psiSteamEnthalpy SteamEnthalpyValue)
      ;; Here we ignore integrated mode ...
      ;; (SETQ BoilerlASteamDrumLevelIndicator 0.0)
      ;; (SETQ BoilerlBSteamDrumLevelIndicator 0.0)
    (COND (150lbSteamEnable?
           (SETQ 150lbAuxSteamGagel 150.0)
           (SETQ 150lbAuxSteamGage2 150.0)
           (SETQ DistillerSteamGage 15.0)
           (SETQ LOPurifierHeaterSteamGage 0.0)
           (SETQ Heater//SteamHeaderPathEnabled NIL)
           (COND ((OR (NOT LOPurifierHeaterSteam)
                      (< LOPurifierValveSteam .1))
                  (SETQ LOPurifierHeaterSteamGage 0.0))
                 (EL906                    ; Whatever that is!
                  (SETQ LOPurifierHeaterSteamGage
                        (*$ .67
                            150lbAuxSteamGagel
                            (-$ LOPurifierValveSteam .1))))
                 (T
                  (SETQ LOPurifierHeaterSteamGage 150lbAuxSteamGagel)))
           (SETQ 150lbSteamPres 150lbAuxSteamGagel)
           (SETQ 150lbSteamAvailable DistillerSteam)
           (SETQ 150lbSteamEnable GlandSealSteam))
          (T
           (SETQ DistillerSteamGage 0.0)
           (SETQ LOPurifierHeaterSteamGage 0.0)
           (SETQ 150lbSteamPres 0.0)
           (SETQ 150lbSteamAvailable NIL)
           (SETQ 150lbSteamEnable NIL)))
    ())


(defun 150psiSteamInit ()
    ; 150psiSteam enabled to GlandSeal, Distiller and LOPurifierHeater.
    (setq GlandSealSteam (select-ic '(nil nil t t)))
    (setq DistillerSteam (select-ic '(nil t t t)))
    (setq LOPurifierHeaterSteam (select-ic '(nil nil nil nil)))

    (setq 150lbSteamEnable? (select-ic '(nil t t t)))   ; instructor knob
    ; HeaterSteamInletValveLift is also initialized in LOTemp.
    (setq HeaterSteamInletValveLift (select-ic '(0.0 0.0 0.0 0.0)))
```

```
    (SETQ SteamEnthalpyValue 1200.0)
    (SETQ LOPurifierValveSteam 0.0)
    (SETQ EL906 NIL))

(version)
```

B.2 AirEjector

```
(declare (unspecial temp) (flonum temp))

(defun MAEandGECondRun ()
  (PROG (temp)
      (SETQ AirEjectorAuralCue NIL)
      (SETQ StagelUnitlAEOperating T)
      (SETQ StagelUnit2AEOperating T)
      (SETQ Stage2UnitlAEOperating T)
      (SETQ Stage2Unit2AEOperating T)
      (COND (SteamInletValve (SETQ DownstreamValvePres 150lbSteamPres))
            (T               (SETQ DownstreamValvePres 0.0)))
      (COND ((NOT (OR (< DownstreamValvePres 150psiSteamMinValue)
                      CasualtyGlandSealLoss
                      (> CoolingWaterTemp AirEjectorBoilingTemp)))
             ; Assuming EJK1 in code is really ESK1.
             ; This is code up to tag 3506.
             (COND ((AND UnitlAirSuctionValvel UnitlAirSuctionValve2)
                    (SETQ UnitlSuctionOn T))
                   (T
                    (SETQ UnitlSuctionOn NIL)))
             (COND (UnitlAirSuctionValve2 (SETQ Stage2UnitlAELinedUp T))
                   (T                     (SETQ Stage2UnitlAELinedUp NIL)))
             (COND ((AND Stage2UnitlAELinedUp
                         UnitlAirSuctionValvel
                         (NOT CasualtyStage2UnitlAEFail))
                    (SETQ UnitlAEInoperative NIL))
                   (T
                    (SETQ UnitlAEInoperative T)))
             (COND ((AND StagelUnitlAELinedUp
                         (NOT CasualtyStagelUnitlAEFail))
                    (SETQ UnitlAEOperating T))
                   (T
                    (SETQ UnitlAEOperating NIL)))
             ;; Now repeat for IE=2, that is, unit 2
             (COND ((AND Unit2AirSuctionValvel Unit2AirSuctionValve2)
                    ;;  Assuming EJQ114 is really EJA114.
                    (SETQ Unit2SuctionOn T))
                   (T
                    (SETQ Unit2SuctionOn NIL)))
             (COND (Unit2AirSuctionValve2 (SETQ Stage2Unit2AELinedUp T))
                   (T                     (SETQ Stage2Unit2AELinedUP NIL)))
```

```
        (COND ((AND Stage2Unit2AELinedUp
                    Unit2AirSuctionValve1
                    (NOT CasualtyStage2Unit2AEFail))
               (SETQ Unit2AEInoperative NIL))
              (T
               (SETQ Unit2AEInoperative T)))
        (COND ((AND Stage1Unit2AELinedUp
                    (NOT CasualtyStage1Unit2AEFail))
               (SETQ Unit2AEOperating T))
              (T
               (SETQ Unit2AEOperating NIL)))
        ; After DO loop at tag 3510
        (COND ((OR Unit1SuctionOn Unit2SuctionOn)
               (SETQ AirEjectorAuralCue T))
              (T
               (SETQ AirEjectorAuralCue NIL)))
        (COND (CondensatePumpGlandSealLoss
               (SETQ CasualtyPresAmplitude 3.0)
               (SETQ CasualtyTC     ; VDT = reciprocal of DeltaT
                     (*$ Stage1FailureTC (//$ 1.0 DeltaT))))
              ((OR Unit1AEOperating Unit2AEOperating)
               (SETQ CasualtyPresAmplitude 0.0)
               (SETQ CasualtyTC (*$ ColdStartFailureTC (//$ 1.0 DeltaT))))
              ((OR Unit1AEInoperative Unit2AEInoperative)
               ; tag 3504
               (SETQ CasualtyPresAmplitude
                     (-$ 3.0 (*$ 2.0354 CondenserSteamPres)))
               (COND ((< CasualtyPresAmplitude 0.0)
                      (SETQ CasualtyPresAmplitude 0.0)))
               (SETQ CasualtyTC
                     (*$ Stage1FailureTC (//$ 1.0 DeltaT))))))
       (T
        (SETQ Stage2Unit1AELinedUp NIL)
        ; tag 3506
        (SETQ Stage2Unit2AELinedUp NIL)
        (COND (CasualtyGlandSealLoss
               (SETQ temp (ToFlonum GlandSealFailure)))
              (T
               (SETQ temp 1.0)))
        (SETQ CasualtyPresAmplitude
              (-$ 30.0 (*$ 2.0354 CondenserSteamPres)))
        (COND ((< CasualtyPresAmplitude 0.0)
               (SETQ CasualtyPresAmplitude 0.0)))
        (SETQ CasualtyTC
              (*$ Stage2FailureTC (//$ 1.0 DeltaT) temp))))
(SETQ CondenserAirPres
      (//$ 1.0 (*$ (+$ CasualtyTC 1.0)
                   (+$ CasualtyPresAmplitude
                       (*$ CasualtyTC CondenserAirPres)))))
(COND (GlandExhaustEjectorValve (SETQ GlandExhaustEjector 0.167))
      (T                        (SETQ GlandExhaustEjector 0.0)))
(SETQ SecondStageUnitsOnCount
      (COND ((AND Stage2Unit1AELinedUp Stage2Unit2AELinedUp) 2.0)
            ((XOR Stage2Unit1AELinedUp Stage2Unit2AELinedUp) 1.0)
```

```
                (T 0.0)))
      (COND ((> EJFRV 0.0) (SETQ EJKFX EJKRX))
            (T             (SETQ EJKFX 1.0)))
      (SETQ temp (*$ CondensateFlow EJKFX))
      (COND ((> temp 2.0) (SETQ temp 2.0)))
      (SETQ a (*$ GlandExhaustEjector .133333))
      (SETQ temp (+$ HotwellTemp
                     (*$ (//$ 7.31 temp)
                         (+$ (*$ (-$ 150psiSteamEnthalpy 180.07)
                                 (+$ (*$ SecondStageUnitsOnCount .2705) a))
                             (*$ (-$ GlandSteamEnthalpy 180.07)
                                 GlandSteamFlow)))))
      (COND ((> temp 240.0) (SETQ temp 240.0)))
      (SETQ CoolingWaterTemp (//$ (+$ (*$ CoolingWaterTemp TempTC)
                                      (*$ temp DeltaT))
                                  (+$ TempTC DeltaT)))
      (COND ((AND (> GlandManifoldSteamPres 0.0)
                  (< CondensateFlow 5.0))
             (Malop 4 bit7)))))


(defun MAEandGECondInit ()
    (setq Stage1Unit2AirSuctionValve (select-ic '(nil nil nil nil)))
    (setq Stage1Unit1AirSuctionValve (select-ic '(nil nil t t)))
    (setq Unit2AirSuctionValve2 (select-ic '(nil nil nil nil)))
    (setq Unit1AirSuctionValve2 (select-ic '(nil nil t t)))
    (setq Unit2AirSuctionValve1 (select-ic '(nil nil nil nil)))
    (setq Unit1AirSuctionValve1 (select-ic '(nil nil t t)))
    (setq SteamInletValve (select-ic '(nil nil t t)))
    (setq GlandExhaustEjectorValve (select-ic '(nil nil t t)))

    (SETQ GlandSteamFlow 0.0)
    (SETQ GlandSealFailure NIL)
    (SETQ GlandSteamEnthalpy 0.0)
    (SETQ EJFRV 0.0)

    (SETQ AirEjectorAuralCue NIL)
    (SETQ CasualtyGlandSteamLoss NIL)
    (SETQ CasualtyStage2Unit2AEFail NIL)
    (SETQ CasualtyStage2Unit1AEFail NIL)
    (SETQ CasualtyStage1Unit2AEFail NIL)
    (SETQ CasualtyStage1Unit1AEFail NIL)
    (SETQ CasualtyGlandSealLoss NIL)

    (SETQ DownstreamValvePres 0.0)
    (SETQ Unit1AEOperating NIL)
    (SETQ Unit1AEInoperative T)
    (SETQ Unit2AEOperating NIL)
    (SETQ Unit2AEInoperative T)
    (SETQ Stage1Unit1AELinedUp NIL)
    (SETQ Stage1Unit2AELinedUp NIL)
    (SETQ Stage2Unit1AELinedUp NIL)
    (SETQ Stage2Unit2AELinedUp NIL)
```

```
    (SETQ CasualtyPresAmplitude 0.0)
    (SETQ CasualtyTC 0.0)
    (SETQ CondenserAirPres 0.0)
    (SETQ GlandExhaustEjector 0.0)
    (SETQ CoolingWaterTemp 70.0)
    (SETQ CondensateFlow 0.0)
    (SETQ GlandSealFailureTC .3)
    (SETQ AirEjectorBoilingTemp 200.0)
    (SETQ TempTC 30.0)
    (SETQ 150psiSteamMinValue 100.0)
    (SETQ ColdStartFailureTC 20.0)
    (SETQ Stage1FailureTC 30.0)
    (SETQ Stage2FailureTC 300.0)
    ())

(version)
```

B.3 AuxExhaust

```
(defun AuxExhaustRun ()
    (SETQ MCUnloadingValveLineupStatus
          InstructorEnableMainCondenserUnloading)
    (SETQ AuxExhaustHeaderStopValve
          InstructorEnableMainCondenserUnloading)
    (SETQ AuxExhaustPres
          IndependentModeAuxExhaustPres)
    (SETQ MainCondenserSteamEnthalpy
          IndependentModeAuxExhaustEnthalpy)
    (SETQ EXWUCM
          AuxExhaustUnloadingWeightFlow)
    (SETQ UnloadingFlowDPEnabled
          InstructorEnableMainCondenserUnloading)
    ; tag 3401
    (SETQ MCUnloadingValveFlowPathEnabled
          (AND MainCondenserUnloadingStopValve
               AuxExhaustHeaderStopValve))
    (SETQ MCUVMMWeightFLow 0.0)
    (SETQ MCUnloadingValveStatus
          (COND (CasualtyUnloadingValveFailClosed NIL)
                (T MCUnloadingValveLineupStatus)))
    (SETQ DPAuxExhaustDemand
          (COND (DPAuxExhaustEnabled .67)
                (T 0.0)))
    (SETQ MCUnloadingValveCommandedValveLift
          (*$ .09473 EXWUCM))
    (COND ((NULL MCUnloadingValveStatus)
           (SETQ EXWUCM 0.0)
           (SETQ MCUnloadingValveCommandedValveLift 0.0)))
    (COND ((NULL MCUnloadingValveFlowPathEnabled)
           (SETQ EXWUCM 0.0))
```

```
        ((NULL MCUnloadingValveManualMode)
         (SETQ MCUVMMWeightFlow
               (*$ MCUnloadingValveManOverrideValveLift 10.5556))
         (SETQ EXWUCM 0.0)))          ; tag 3403
         (COND ((= AuxExhaustPres 0.0)
                (SETQ MCUVMMWeightFlow 0.0)))
         (SETQ MainCondenserSteamFlow
               (+$ MCUVMMWeightFLow EXWUCM))
         (SETQ AuxExhaustSupplyPres
               (+$ (COND (AuxExhaustHeaderStopValve AuxExhaustPres)
                         (T 0.0))
                   (COND ((AND UnloadingFlowDPEnabled
                               MainCondenserUnloadingStopValve)
                          (*$ .1895 (*$ MainCondenserSteamFlow
                                        DPAuxExhaustDemand)))
                         (T 0.0))))
         (SETQ AuxExhaustUnloadingPres
               (COND
                 (MainCondenserUnloadingStopValve AuxExhaustSupplyPres)
                 (T 0.0)))
         ())


(defun AuxExhaustInit ()
    (setq MCUnloadingValveManOverrideValveLift (select-ic '(0.0 0.0 0.0 0.0)))
    (setq MCUnloadingValveManualMode (select-ic '(nil nil nil nil)))
    (setq InstructorEnableMainCondenserUnloading (select-ic '(nil nil t t)))
    (setq MainCondenserUnloadingStopValve (select-ic '(nil nil t t)))

    (SETQ DPAuxExhaustEnabled NIL)
    (SETQ MCUnloadingValveFlowPathEnabled NIL)
    (SETQ AuxExhaustHeaderStopValve NIL)
    (SETQ CasualtyUnloadingValveFailClosed NIL)

    (SETQ MainCondenserSteamFlow 0.0)
    (SETQ IndependentModeAuxExhaustEnthalpy 1163.0)
    (SETQ IndependentModeAuxExhaustPres 16.0)
    (SETQ AuxExhaustUnloadingWeightFlow 5.0)
    (SETQ MCUnloadingValveCommandedValveLift 0.0))

(version)
```

B.4 ControlAir

```
(defun ControlAirRun ()
    (COND (ControlAirOn (SETQ ControlAirGage1 130.0)
                        (SETQ ControlAirGage2 100.0))
          (T (SETQ ControlAirGage1 0.0)
             (SETQ ControlAirGage2 0.0)))
    (COND ((NOT (< ControlAirGage1 20.0))
           (COND (AuxExhaustToDistilPlant (SETQ AuxExhaustToDPCAPres 20.0))
```

```
                (t                          (SETQ AuxExhaustToDPCAPres 0.0)))
            (COND (LOServiceUnloading (SETQ LOServiceUnloadingValve 20.0))
                  (t                  (SETQ LOServiceUnloadingValve 0.0)))
            (COND (MTGlandSealUnloading (SETQ SupplyValveCAPres 20.0))
                  (t                    (SETQ SupplyValveCAPres 0.0)))
            (COND (MTGlandSealSupply (SETQ UnloadingValveCAPres 20.0))
                  (t                 (SETQ UnloadingValveCAPres 0.0))))
           (T
            (SETQ AuxExhaustToDPCAPres 0.0)
            (SETQ LOServiceUnloadingValve 0.0)
            (SETQ SupplyValveCAPres 0.0)
            (SETQ UnloadingValveCAPres 0.0)))
    (COND ((OR (< ControlAirGage2 60.0)
               (NOT SteamControlAheadGuarding)
               (NOT AheadGuarding))
           (SETQ ResetAheadGuardingMotorEnable NIL))
          (T
           (SETQ ResetAheadGuardingMotorEnable T)))
    ())


(defun ControlAirInit ()
    (setq AheadGuarding (select-ic '(nil nil t t)))
    ; AheadGuarding is also in MainSteam and MainTurbine.
    (setq SteamControlAheadGuarding (select-ic '(nil nil nil nil)))
    (setq AuxExhaustToDistilPlant (select-ic '(nil t t t)))
    (setq MTGlandSealUnloading (select-ic '(nil nil t t)))
    (setq MTGlandSealSupply (select-ic '(nil nil t t)))
    (setq LOServiceUnloading (select-ic '(t t t t)))
    (setq ControlAirOn (select-ic '(nil t t t))))

(version)
```

B.5 Distiller

```
(version)

(defun DistillerRun ()
    (SETQ EngineRoomSeaWaterTemp
          (+$ 30.0 (*$ 55.0 SeaWaterTempInstructorKnob)))
    (COND ((AND FeedWaterSeaChestValve FeedPumpMotorOn FeedWaterPumpOutletValve)
           ; tag 4308
           (SETQ FeedwaterStrainerFeedSuction 14.0)
           (SETQ FeedwaterPresToStage1 50.0)
           (SETQ FeedwaterInletTemp EngineRoomSeaWaterTemp)
           (SETQ FeedwaterPumpOutletPres 59.0))
          (T
           (COND ((AND FeedWaterSeaChestValve FeedPumpMotorOn)
                  (SETQ FeedwaterPumpOutletPres 60.0))
                 (T
                  (SETQ FeedwaterPumpOutletPres 14.7)))
```

```
        (SETQ FeedwaterStrainerFeedSuction 14.7)
        (SETQ FeedwaterPresToStage1 14.7)
        (SETQ FeedwaterInletTemp 75.0)))      ; tag 4310
(COND (DistillerBrineDischargeOn    ; tag 4312
        (COND (BrineOverboardValve
               (SETQ BrineEductorSound T)
               (SETQ BrineOverboardDischargePres 12.2))
              (T
               (SETQ BrineEductorSound NIL)
               (SETQ BrineOverboardDischargePres 40.0))))
      (T
        (COND (BrineOverboardValve (Malop 6 bit23)))
        (SETQ BrineEductorSound NIL)
        (SETQ BrineOverboardDischargePres 14.7)))
; tag 4352
(COND ((AND 150lbSteamAvailable (NULL EBCSTM))
        (SETQ AuxSteamToDAEPres 150lbSteamPres)
        (COND ((NOT FeedPumpMotorOn) (Malop 6 bit23)))
        (COND (Stage1AESteamValve
               (SETQ DPresInput T)
               (SETQ DPresResponseGain PresRiseResponseGain))
              (T
               (SETQ DPresInput NIL)
               (SETQ DPresResponseGain PresDecayResponseGain))))
      (T
        (SETQ AuxSteamToDAEPres 150lbSteamPres)
        (SETQ DPresInput NIL)
        (SETQ DPresResponseGain PresDecayResponseGain)))
; tag 4315
(SETQ DPresResponseError (-$ (COND (DPresInput 1.0) (T 0.0))
                             DPresResponseOutput))
(SETQ DPresResponseOutput (+$ DPresResponseOutput
                              (*$ DPresResponseGain
                                  DPresResponseError
                                  DeltaT)))
(SETQ Stage1ShellPres (-$ 14.7 (*$ 13.95 DPresResponseOutput)))
(SETQ STage2ShellPres Stage1ShellPres)

; tag 4353, 3 on flowchart

(COND ((< AuxExhaustToDPCAPres 20.0)
        (SETQ AirPilotInput NIL)
        (SETQ 3psigSteamControlValve 0.0)
        (SETQ 3to15psigAirPilotOutputPres 0.0)
        (SETQ DPAuxExhaustEnabled NIL)
        (SETQ AuxSteamToOrificePres 14.7)
        (SETQ DTempInput NIL)
        (SETQ TempResponseGain TempDecayResponseGain))
      (T
        (SETQ AirPilotInput T)
        (COND ((AND AuxExhaustSteamValve AuxExhaustSteamValve1
                    AuxExhaustSteamValve2 AuxExhaustSteamValve3
                    (OR (> AuxExhaustSupplyPres 3.0)
                        (= DistillerSteamGage 15.0)))
```

```
                (COND ((OR (NOT FeedPumpMotorOn)
                           (> StagelShellPres 2.45))
                       (Malop 6 bit23)))
                (SETQ 3psigSteamControlValve .25)
                (SETQ 3tol5psigAirPilotOutputPres (+$ 3.0 (*$ .25 12.0)))
                (SETQ DPAuxExhaustEnabled T)
                (SETQ AuxSteamToOrificePres 17.7)
                (SETQ DTempInput T)
                (SETQ TempResponseGain .02))
               (T
                (SETQ 3psigSteamControlValve 1.0)
                (SETQ 3tol5psigAirPilotOutputPres 15.0)
                (SETQ DPAuxExhaustEnabled NIL)
                (SETQ AuxSteamToOrificePres 14.7)
                (SETQ DTempInput NIL)
                (SETQ TempResponseGain TempDecayResponseGain)))))

; Tag 4354 in code, 4 in flowchart
; (Well, actually, this stuff is not in the flowchart.)

(COND ((AND FeedPumpMotorOn FeedWaterPumpOutletValve
            (>  FeedWaterTempControlValve 0.0))
       (COND (EBSC3F (SETQ EBSC3F NIL)
                     (SETQ EBSINS3 1.5))
             (T (COND (DPresInput (SETQ EBSC3T EBSCOF))
                      (T (SETQ EBSC3T EBSCNG)))
                (SETQ EBSC3C (MAX (-$ (*$ StagelShellPres .1196) .258)
                                  .035))
                (SETQ EBSINS3 (//$ (+$ (*$ EBSINS3  EBSC3T)
                                       (*$ EBSC3C  DeltaT))
                                   (+$ EBSC3T DeltaT))))))
       (T                                ; tag 4319
        (SETQ EBSC3F T)))
(COND ((AND DistillatePumpMotorOn DistillatePumpOutletValve)
       (SETQ EBSINS6 EBSINS3)))
(SETQ TempResponseError (-$ (COND (DTempInput 1.0) (T 0.0))
                            TempResponseOutput))
(SETQ TempResponseOutput
      (+$ TempResponseOutput
          (*$ TempResponseGain TempResponseError DeltaT)))
(SETQ DTemp3 (+$ 33.5 (*$ FeedwaterInletTemp .3491)))
(SETQ DTemp4 (-$ 125.0 (*$ FeedWaterTempControlValve 100.0)))
(SETQ DTemp5 (+$ 41.0 (*$ FeedwaterInletTemp .3175)))
(SETQ DTemp6 (-$ (*$ .6333 FeedwaterInletTemp) 12.0))
(SETQ DTemp7 (-$ (*$ .7895 FeedwaterInletTemp) 47.4))
(SETQ DTemp8 (-$ (*$ .6666 FeedwaterInletTemp) 19.6))
(SETQ FeedwaterInAfterCondenserTemp
      (+$ (*$ DTemp3 TempResponseOutput) 75.0))
(SETQ FeedwaterOutAfterCondenserTemp
      (+$ (*$ DTemp4 TempResponseOutput) 75.0))
(SETQ StagelShellTemp (+$ (*$ DTemp5 TempResponseOutput) 75.0))
(SETQ Stage2ShellTemp (+$ (*$ DTemp6 TempResponseOutput) 75.0))
(SETQ FeedwaterInterstageTemp (+$ (*$ DTemp8 TempResponseOutput) 75.0))
(SETQ SteamToSWHeater    (+$ 75.0 (*$ 130.0 TempResponseOutput)))
```

```
(SETQ SWHeaterShellTemp (+$ 75.0 (*$ 123.0 TempResponseOutput)))
(SETQ StagelShellPres (+$ StagelShellPres
                          (*$ 2.36 TempResponseOutput)))
(SETQ Stage2ShellPres (+$ Stage2ShellPres
                          (*$  .8  TempResponseOutput)))
(COND ((AND SWHeaterDrainPumpMotorOn HeaterDrainRegJackSpindleOpen)
       (SETQ SWHeaterDrainPumpSound T)
       (SETQ CondensateOutSWHeaterTemp SWHeaterShellTemp)
       (SETQ SWHeaterDrainPumpDischargePres 55.0)
       (SETQ HeaterLevelError (-$ HeaterLevelInput HeaterLevelOutput))
       (SETQ HeaterLevelOutput (+$ HeaterLevelOutput
                                   (*$ .02 HeaterLevelError DeltaT)))
       (COND ((NULL SWHeaterDrainPumpSealValve) (Malop 6 bit23)))
       (COND (CondensateSystemValve
              (SETQ CondensateToCondensateSystem T)
              (COND ((OR CondensateToBilgeValve SteamDrainSystemValve)
                     (Malop 6 bit23)))
              (SETQ CondensateToBilge NIL)
              (SETQ CondensateToSteamDrainSystem NIL))
             (T
              (SETQ CondensateToCondensateSystem NIL)
              (SETQ CondensateToBilge CondensateToBilgeValve)
              (SETQ CondensateToSteamDrainSystem SteamDrainSystemValve)
              (COND ((NOR CondensateToBilgeValve SteamDrainSystemValve)
                     (Malop 6 bit23))))))
      (T
       (COND (SWHeaterDrainPumpMotorOn
              (SETQ SWHeaterDrainPumpSound T)
              (SETQ SWHeaterDrainPumpDischargePres 75.0))
             (T
              (SETQ SWHeaterDrainPumpSound NIL)
              (SETQ SWHeaterDrainPumpDischargePres 14.7)))
       (SETQ CondensateOutSWHeaterTemp 75.0)
       (COND ((< FeedwaterOutAfterCondenserTemp FeedwaterOutRefTemp)
              (SETQ HeaterLevelGain HeaterLevelFallGain))
             (T
              (SETQ HeaterLevelGain HeaterLevelRiseGain)))
       (SETQ HeaterLevelOutput
             (MIN 1.0 (+$ HeaterLevelOutput
                          (*$ HeaterLevelGain DeltaT))))
       (SETQ CondensateToCondensateSystem NIL)
       (SETQ CondensateToBilge NIL)
       (SETQ CondensateToSteamDrainSystem NIL)))
(SETQ HeaterLevelOutput (MAX HeaterLevelOutput 0.0))

; tag 4356, 6 in flowchart

(COND ((AND DistillatePumpMotorOn DistillatePumpOutletValve)
       (SETQ DistillatePumpSound T)
       (SETQ DistillateLevelInput (-$ DistillateZeroPresLevel
                                      (*$ DistillateLevelOutputGain
                                          Stage2ShellPres)))
       (SETQ DistillateLevelError (-$ DistillateLevelInput
                                      DistillateLevelOutput))
```

```
          (SETQ DistillateLevelOutput (+$ DistillateLevelOutput
                                          (*$ DistillateLevelOutputGain
                                              DistillateLevelError
                                              DeltaT)))
          (COND ((NULL DistillatePumpSealValve) (Malop 6 bit23)))
          (SETQ DistillatePumpDischargePres 37.0)
          (SETQ FinalDistillateTemp (+$ 75.0
                                        (*$ DTemp7 TempResponseOutput)))
          (SETQ DistillateAvailable T))
         (T
          (COND (DistillatePumpMotorOn
                 (SETQ DistillatePumpSound T)
                 (SETQ DistillatePumpDischargePres 60.0))
                (T
                 (SETQ DistillatePumpSound NIL)
                 (SETQ DistillatePumpDischargePres 14.7)))
          (SETQ FinalDistillateTemp 75.0)
          (SETQ DistillateAvailable NIL)
          (COND ((> Stage2ShellPres Stage2RefPres)
                 (SETQ DistillateLevelGain DistillateLevelFallGain))
                (T
                 (SETQ DistillateLevelGain DistillateLevelRiseGain)))
          (SETQ DistillateLevelOutput
                (FBOUNDS 0.0
                         (+$ DistillateLevelOutput
                             (*$ DistillateLevelGain DeltaT))
                         1.0))))
    (COND ((> HeaterLevelOutput CondensateManometerBottomLevel)
           (SETQ CondensateManometerLevel
                 (//$ (-$ HeaterLevelOutput CondensateManometerBottomLevel)
                      (-$ 1.0 CondensateManometerBottomLevel))))
          (T
           (SETQ CondensateManometerLevel 0.0)))
    (COND ((> DistillateLevelOutput DistillateManometerBottomLevel)
           (SETQ DistillateManometerLevel
                 (//$ (-$ DistillateLevelOutput
                          DistillateManometerBottomLevel)
                      (-$ 1.0
                          DistillateManometerBottomLevel))))
          (T
           (SETQ DistillateManometerLevel 0.0)))
    (COND ((AND CondensateToCondensateSystem
                (> CondensateSystemLoadPres 60.0))
           (SETQ CondensateToCondensateSystem NIL)
           (SETQ SWHeaterDrainPumpDischargePres 75.0)))
    ())


(defun DistillerInit ()
    (setq SteamDrainSystemValve (select-ic '(nil nil nil nil)))
    (setq CondensateToBilgeValve (select-ic '(nil nil nil nil)))
    (setq CondensateSystemValve (select-ic '(nil t t t)))
    (setq BrineOverboardValve (select-ic '(nil t t t)))
    (setq SWHeaterDrainPumpSealValve (select-ic '(nil t t t)))
```

```
    (setq DistillatePumpSealValve (select-ic '(nil t t t)))
    (setq StagelAESteamValve (select-ic '(nil t t t)))
    (setq DistillatePumpOutletValve (select-ic '(nil t t t)))
    (setq FeedWaterPumpOutletValve (select-ic '(nil t t t)))
    (setq FeedWaterSeaChestValve (select-ic '(nil t t t)))
    (setq AuxExhaustSteamValve3 (select-ic '(nil t t t)))
    (setq AuxExhaustSteamValve2 (select-ic '(nil t t t)))
    (setq AuxExhaustSteamValvel (select-ic '(nil t t t)))
    (setq AuxExhaustSteamValve (select-ic '(nil t t t)))
    (setq HeaterDrainRegJackSpindleOpen (select-ic '(nil t t t)))
    (setq FeedWaterTempControlValve (select-ic '(0.0 1.0 1.0 1.0)))
    (setq SeaWaterTempInstructorKnob
          (select-ic '(0.8181 0.8181 0.8181 0.8181)))

    (SETQ Stage2ShellPres 0.0)
    (SETQ StagelShellPres 0.0)
    (SETQ DPresResponseOutput 0.0)
    (SETQ TempResponseOutput 0.0)
    (SETQ DistillateLevelOutput 0.0)
    (SETQ HeaterLevelOutput 0.0)
    (SETQ EBSINS3 0.0)
    (SETQ EBSC3F NIL)
    (SETQ EBSCNG 0.0)
    (SETQ EBCSTM NIL)                    ; Unknown variable

    ;; Model Constants
    (SETQ PresDecayResponseGain .0025)
    (SETQ PresRiseResponseGain .01)
    (SETQ EBMC15 .25)
    (SETQ TempRiseResponseGain .02)
    (SETQ TempDecayResponseGain .01)
    (SETQ EBMC31 12.2)
    (SETQ HeaterLevelRiseGain .005)
    (SETQ HeaterLevelFallGain -.0025)
    (SETQ HeaterLevelInput .75)
    (SETQ CondensateManometerBottomLevel .5)
    (SETQ Stage2RefPres 7.0)
    (SETQ DistillateLevelRiseGain .004)
    (SETQ DistillateLevelFallGain -.002)
    (SETQ DistillateZeroPresLevel 1.0)
    (SETQ DistillateManometerBottomLevel .4)
    (SETQ FeedwaterOutRefTemp 125.0)
    (SETQ CondensateLevelGain .02)
    (SETQ DistillateLevelInputGain .1)
    (SETQ DistillateLevelOutputGain .01)
    (SETQ EBMC30 14.0)
    ())


(defun DistillerStatus ()
   (StatusPrint DistillerStatusVars))

(setq DistillerStatusVars
      '(EngineRoomSeaWaterTemp FinalDistillateTemp
```

```
        Stage1ShellPres Stage1ShellTemp Stage2ShellPres Stage2ShellTemp
        CondensateManometerLevel DistillateManometerLevel))
```

B.6 DPumps

```
(defun DPumpsRun ()
    (COND ((OR (NOT FeedPumpPowerOn) FeedPumpStop)
           (SETQ FeedPumpMotorOn NIL))
          (FeedPumpStart
           (SETQ FeedPumpMotorOn T)))
    (COND ((OR (NOT DistillatePumpPowerOn) DistillatePumpStop)
           (SETQ DistillatePumpMotorOn NIL))
          (DistillatePumpStart
           (SETQ DistillatePumpMotorOn T)))
    (COND ((OR (NOT SWHeaterDrainPumpPowerOn) SWHeaterDrainPumpStop)
           (SETQ SWHeaterDrainPumpMotorOn NIL))
          (SWHeaterDrainPumpStart
           (SETQ SWHeaterDrainPumpMotorOn T)))
    ())


(defun DPumpsInit ()
    (setq SWHeaterDrainPumpStart (select-ic '(nil nil nil nil)))
    (setq SWHeaterDrainPumpStop (select-ic '(nil nil nil nil)))
    (setq DistillatePumpStart (select-ic '(nil nil nil nil)))
    (setq DistillatePumpStop (select-ic '(nil nil nil nil)))
    (setq FeedPumpStart (select-ic '(nil nil nil nil)))
    (setq FeedPumpStop (select-ic '(nil nil nil nil)))
    (SETQ FeedPumpPowerOn NIL)
    (SETQ FeedPumpMotorOn NIL)
    (SETQ DistillatePumpPowerOn NIL)
    (SETQ DistillatePumpMotorOn NIL)
    (SETQ SWHeaterDrainPumpPowerOn NIL)
    (SETQ SWHeaterDrainPumpMotorOn NIL))

(defun TurnOffDPumps ()
    (SETQ DistillatePumpStop T)
    (SETQ DistillatePumpStart NIL)
    (SETQ FeedPumpStop T)
    (SETQ FeedPumpStart NIL)
    (SETQ SWHeaterDrainPumpStop T)
    (SETQ SWHeaterDrainPumpStart NIL))

(defun TurnOnDPumps ()
    (SETQ DistillatePumpStop NIL)
    (SETQ DistillatePumpStart T)
    (SETQ FeedPumpStop NIL)
    (SETQ FeedPumpStart T)
    (SETQ SWHeaterDrainPumpStop NIL)
    (SETQ SWHeaterDrainPumpStart T))
```

```
(defun DPumpsStatus ()
   (statusprint DPumpsStatusVars))

(setq DPumpsStatusVars '(DistillatePumpMotorOn
                         FeedPumpMotorOn
                         SWHeaterDrainPumpMotorOn))

(version)
```

B.7 Electric

```
(declare (unspecial d) (flonum d))

(defun ElectDistRun ()
 (PROG (d)                                    ; EELD4
   (SETQ CondPump1APowerOn T)
   (SETQ CondPump1BPowerOn T)
   (SETQ LOPumpsPowerOn T)
   (SETQ d (+$ (*$ 26.0 (FSUM-VARS CondPump1APowerOn
                                   CondPump1BPowerOn))
               (*$ 51.7 (FSUM-VARS (LOPumpHighIndicator 1)
                                   (LOPumpHighIndicator 2)))
               (*$ 34.8 (FSUM-VARS (LOPumpLowIndicator 1)
                                   (LOPumpLowIndicator 2)))
               (COND ((NOT LightingEnable) (EELD 3))
                     (T 0.0))
               68.0))
   (store (EELDN 4) d)
   (store (EELDA 4) 0.0)
   (store (EELD 1) (COND (FirePump3MotorOn 50.0)
                         (T 0.0)))
   (store (EELD 3) (COND (CircPumpHighSpeedIndicator 60.0)
                         (CircPumpLowSpeedIndicator 30.0)
                         (T 0.0)))
   (for i from 1 to 3
      do (store (PowerEnableAft i)
                (COND ((AftDevices i) NIL)      ; T = closed
                      (T (AftLamps i))))
         (store (PowerEnableCentral i)
                (COND ((CentralDevices i) NIL)
                      (T (CentralLamps i))))
         (store (EELDN i)
                (COND ((PowerEnableCentral i) (EELD i))
                      (T 0.0)))
         (store (EELDA i)
                (COND ((PowerEnableAft i) (EELD i))
                      (T 0.0)))
      finally (SETQ AuralCueEnableVentFans (OR (PowerEnableCentral 2)
                                               (PowerEnableAft 2))))
```

```
    (SETQ FirePump3PowerOn (OR (PowerEnableCentral 1)
                               (PowerEnableAft 1)))
    (SETQ CircPumpPowerOn (OR (PowerEnableCentral 3)
                              (PowerEnableAft 3)))
    (SETQ FeedPumpPowerOn T)              ; distiller
    (SETQ DistillatePumpPowerOn T)
    (SETQ SWHeaterDrainPumpPowerOn T)     ; distiller
    (SETQ LOPurifierPowerOn T)
    (SETQ JMPowerOn T)
    (store (EELDN 5)
           (+$ (COND (FeedPumpMotorOn 12.7)      ; distiller
                     (T 0.0))
               (COND (DistillatePumpMotorOn 3.85)
                     (T 0.0))
               (COND (SWHeaterDrainPumpMotorOn 7.0)      ; distiller
                     (T 0.0))
               (COND (LOPurifierMotorOn 2.5)
                     (T 0.0))
               (COND ((NULL JackingMotorStatus) 0.0)
                     (T 5.0))))
    (SETQ TotalLoadCentral (for i from 1 to 5 sum (EELDN i)))
    (SETQ TotalLoadAft     (for i from 1 to 4 sum (EELDA i)))))


(defun ElectDistInit ()
    ; These variables have arrays as their values.
    (FILLARRAY EELD  '(0.0))
    (FILLARRAY EELDN '(0.0))
    (FILLARRAY EELDA '(0.0))
    (FILLARRAY CentralDevices '(()))
    (FILLARRAY AftDevices '(T))
      ;(setq CentralDevices (select-ic '(nil nil nil nil)))
      ;(setq AftDevices (select-ic '(t t t t)))
    (FILLARRAY AftLamps '(()))
    (FILLARRAY CentralLamps '(()))
    (FILLARRAY PowerEnableAft '(()))
    (FILLARRAY PowerEnableCentral '(()))

    (store (EELD 2) 20.0)                 ; model constant (amps)
    (SETQ TotalLoadAft 0.0)
    (SETQ TotalLoadCentral 0.0)
    (SETQ LightingEnable NIL)
    ())

(defun TurnOnElectDist ()
    (for i from 1 to 3 do (store (AftLamps i) T)
                          (store (CentralLamps i) T)))

(defun ElectDistStatus ()
    (statusprint ElectDistStatusVars))

(setq ElectDistStatusVars
      '(TotalLoadCentral TotalLoadAft))
```

```
(version)
```

B.8 FireMain

```
(declare (unspecial a) (flonum a))

(defun FireMainRun ()
 (PROG (a)
   (SETQ MainHeaderAvailable T)
   ; It remains to be seen what parts of this are relevant.
   (WPUMP FirePump3On 1 EWP3SV FirePump3Header
          FirePump3DischargeGage FirePump3SuctionGage
          FirePump3control FirePump3Control1 FirePump3Control2
          FirePump3PowerOn MainHeaderAvailable)
   (COND ((OR MainHeaderAvailable
              (NOT (= FirePump3Header 0.0)))
          (SETQ AuxSWCoolingHeaderOn
                FireMainToAuxSWCooling)
          (SETQ EmergencyCondenserSupplyHeaderOn
                MainCondenserCoolingValve)
          (SETQ DistillerBrineDischargeOn
                FireMainToDistillerBrineEductors)
          (COND (DistillerBrineDischargeOn
                 (SETQ DistillerBrineOverboardGage 125.0))
                (T
                 (SETQ DistillerBrineOverboardGage 0.0)))
          (SETQ a 0.0)
          (COND (AuxSWCoolingHeaderOn (SETQ a (+$ a 1.0))))
          (COND (EmergencyCondenserSupplyHeaderOn (SETQ a (+$ a 1.0))))
          (COND (MainHeaderAvailable
                 (COND (EmergencyCoolingDieseLGenOn (SETQ a (+$ a 1.0))))
                 (COND (AuxMachSWCoolingOn (SETQ a (+$ a 1.0))))))
          (SETQ FireMainPres (-$ 125.0 a)))
         (T
          (SETQ AuxSWCoolingHeaderOn NIL)
          (SETQ EmergencyCondenserSupplyHeaderOn NIL)
          (SETQ DistillerBrineDischargeOn NIL)
          (SETQ DistillerBrineOverboardGage 0.0)
          (SETQ FireMainPres 0.0)))))


(defun FireMainInit ()
   (setq FireMainHoseAdaptorValve (select-ic '(nil nil nil nil)))
   (setq FireMainStopValve3 (select-ic '(nil nil nil nil)))
   (setq FireMainStopValve2 (select-ic '(nil nil nil nil)))
   (setq FireMainStopValve1 (select-ic '(nil nil nil nil)))
   (setq FirePump3Control2 (select-ic '(nil nil nil nil)))
   (setq FirePump3Control1 (select-ic '(nil nil nil nil)))
   (setq FirePump3Control (select-ic '(nil nil nil nil)))
   (setq FirePump3SuctionValve (select-ic '(nil t t t)))
   (setq FireMainToAuxSWCooling (select-ic '(nil t t t)))
```

```
    (setq FireMainToDistillerBrineEductors (select-ic '(nil t   t)))
    (setq MainCondenserCoolingValve (select-ic '(nil t t t)))

    (SETQ AuxMachSWCoolingOn NIL)
    (SETQ EmergencyCoolingDieselGenOn NIL)
    (SETQ DistillerBrineDischargeOn NIL)
    (SETQ MainHeaderAvailable NIL)

    (SETQ EWP3SV 0.0)
    (SETQ FirePump3Header 0.0)
    (SETQ FirePump3DischargeGage 0.0)
    (SETQ FirePump3SuctionGage 0.0)
    (SETQ FirePump3PowerOn NIL)
    (SETQ FirePump3MotorOn NIL)
    ())

;;; Supposedly defined somewhere as a generic water pump model.
(defun WPUMP fexpr (x) x)

(version)
```

B.9 JackingMotor

```
(declare (unspecial temp) (flonum temp))

(defun JackingMotorRun ()
  (PROG (temp)
    (COND ((OR (NOT JMPowerOn) JMStop//Reset)
           (COND (JMStop//Reset (SETQ JMRelayFlag NIL)))
           (SETQ JackingMotorStatus NIL))
          ((NULL JackingMotorStatus)
           (COND (JMStartForward
                  (COND ((NULL JMRelayFlag)
                         (SETQ JackingMotorStatus 'Forward))))
                 (JMStartReverse
                  (COND ((NULL JMRelayFlag)
                         (SETQ JackingMotorStatus 'Reverse)))))))
    ; At A in flowchart, tag 4510 + 1
    (COND ((AND CasualtyInstructorRumble
                (NOT (= ShaftRotationRPM 0.0))
                (NOT (= EKJNM 0.0)))
           (SETQ RumbleSignalAuralCue
                 (FBOUNDS 0.0
                          (+$ JMRumbleCurveIntercept
                              (*$ JMRumbleCurveSlope (ABS ShaftRotationRPM)))
                          10.0)))
          (T
           (SETQ RumbleSignalAuralCue 0.0)))
    ; tag 4526, B in flowchart
    (COND (JMEngage
           (COND ((NULL JMFirstEngageFlag)
```

```
                (COND ((NOT (AND (< (ABS ShaftRotationRPM) EKR)
                                 (NULL JackingMotorStatus)))
                       (SETQ FreezeFlag 3)
                       (FREEZE 2 1))
                      (T
                       (SETQ JMFirstEngageFlag T)))))
          (COND (JMBrake
                 (COND (JackingMotorStatus
                        (SETQ JackingMotorStatus NIL)
                        (SETQ JMRelayFlag T))
                       (T
                        (SETQ RotationOfShaft
                              (//$ (*$ EK2TAU ShaftRotationRPM)
                                   (+$ EK2TAU DeltaT)))
                        (SETQ ShaftRotationRPM RotationOfShaft)
                        (SETQ ShaftRotation (//$ ShaftRotationRPM 60.0)))))
                (T
                 (SETQ temp
                       (+$ ShaftRotationRPM
                           (COND ((EQ JackingMotorStatus 'Forward)  1.0)
                                 ((EQ JackingMotorStatus 'Reverse) -1.0)
                                 (T 0.0))))
                 (SETQ RotationOfShaft
                       (//$ (+$ (*$ JMTC RotationOfShaft)
                                (*$ DeltaT temp))
                            (+$ JMTC DeltaT))))))
         (T                                ; Jacking motor not engaged
          (SETQ RotationOfShaft ShaftRotationRPM)
          (SETQ JMFirstEngageFlag NIL)
          (COND (JMBrake (COND (JackingMotorStatus
                                (SETQ JackingMotorStatus NIL)
                                (SETQ JMRelayFlag T))))
                (JackingMotorStatus (Malop blank)))))
    ; C in flowchart, tag 4553
    (COND (1501bSteamEnable
           (COND ((= RotationOfShaft 0.0)
                  (SETQ JMLapsedTC (+$ JMLapsedTC DeltaT))
                  (COND ((NOT (< JMLapsedTC 180.0))
                         (SETQ CasualtyInstructorRumble 3.0))))
                 (T
                  (COND ((< LOTemp 90.0) (Malop 12 bit23)))
                  (SETQ JMLapsedTC 0.0))))
          (T
           (SETQ JMLapsedTC 0.0)))
    ;; At F in flowchart, tag 4560
    (SETQ JackingGearForward (EQ JackingMotorStatus 'Forward))
    (SETQ JackingGearReverse (EQ JackingMotorStatus 'Reverse))
    (SETQ MainShaftNormalizedRPM (//$ (ABS RotationOfShaft) 240.0))))


(defun JackingMotorInit ()
    (setq JMStartReverse (select-ic '(nil nil nil nil)))
    (setq JMStartForward (select-ic '(nil nil nil nil)))
    (setq JMStop//Reset (select-ic '(nil nil nil nil)))
```

```
    (setq JMBrake (select-ic '(nil nil nil nil)))
    (setq JMEngage (select-ic '(nil nil nil nil)))

    (SETQ JMRelayFlag NIL)
    (SETQ JMPowerOn NIL)
    (SETQ CasualtyInstructorRumble NIL)
    (SETQ JackingMotorStatus NIL)
    (setq JMLapsedTC 0.0)

    ;; Model Constants
    (SETQ JMTC 8.0)
    (SETQ JMRumbleCurveSlope .1)
    (SETQ JMRumbleCurveIntercept -5.0)
    (SETQ EK2TAU 2.0)
    (SETQ EKR .05)
    ())

(defun TurnOffJackingMotor ()
    (SETQ JMStop//Reset T)
    (SETQ JMStartForward NIL))

(defun TurnOnJackingMotor ()
    (SETQ JMStop//Reset NIL)
    (SETQ JMStartForward T))

(defun JackingMotorStatus ()
    (StatusPrint JackingMotorStatusVars))

(setq JackingMotorStatusVars
      '(JackingMotorStatus JMBrake JMRelayFlag ShaftRotation))

(version)
```

B.10 LOPres

```
(version)

(declare (unspecial a b c d e f p v temp) (flonum a b c d e f p v temp))

(defun LOPresRun ()
  (PROG (a b c d e f p v temp)
    (COND ((NOT (= XMODE2 0))     ; Whatever this means - Accelerated Mode?
           (SETQ p AccelMode3)
           (SETQ v AccelMode2)
           (SETQ f AccelMode1))
          (T
           (SETQ p LOTurbineInletPresFlowTC)
           (SETQ v LOUnloadingValveFlowTC)
           (SETQ f LOServicePumpFlowTC)))
    (COND ((> LOSystemCapacity LowSumpLevel)
           (SETQ a (+$ (*$ .334 (-$ MainSumpOilTemp 40.0)) 220.0))
```

```
        (for i from 1 to 2
          do (SETQ temp (COND ((LOPumpHighIndicator i)
                               (COND ((LOPumpLowIndicator i) 3.0)
                                     (T 2.0)))
                              ((LOPumpLowIndicator i) 1.0)
                              (T 0.0)))
             (store (LOServicePumpFlowCommand i)
                    (//$ (+$ (*$ (LOServicePumpFlowCommand i) f)
                             (*$ temp DeltaT))
                         (+$ f DeltaT)))
             (store (LOPumpFlow i)
                    (*$ a (LOServicePumpFlowCommand i))))
        (COND (CasualtyAttachedPumpFail
               (SETQ AttachedLOPumpFlow 0.0))
              (T
               (SETQ AttachedLOPumpFlow
                     (MAX 0.0 (*$ ShaftRotationRPM
                                  (+$ (*$ .00316 (-$ MainSumpOilTemp
                                                     40.0))
                                      1.833)))))))
       (T
        (store (LOPumpFlow 1) 0.0)
        (store (LOPumpFlow 2) 0.0)
        (SETQ AttachedLOPumpFlow 0.0)))
; tag 4602
(SETQ TotalLOPumpFlow (+$ AttachedLOPumpFlow
                          (LOPumpFlow 1)
                          (LOPumpFlow 2)))
(COND ((< TotalLOPumpFlow 25.0)
       (SETQ CasualtyLOCoolerFailure NIL))
      (T
       (SETQ CasualtyLOCoolerFailure (AND LOHeaterValve1
                                          LOHeaterValve2
                                          LOHeaterValve3))))
(SETQ ControlAirLinedUp (= LOServiceUnloadingValve 20.0))
(SETQ SupplyAirV595 ControlAirLinedUp)
(SETQ LOTempCoeff (+$ (*$ .00263 MainSumpOilTemp) .316))
(COND ((OR (NOT ControlAirLinedUp) CasualtyUnloadingValveStuckShut)
       ; tag 4603
       (SETQ IndicatorV595 (COND (CasualtyUnloadingValveStuckShut 1.0)
                                 (T 0.0)))
       (SETQ OutputAirV595 0.0))
      (T
       (COND ((< LOPres 16.0) (SETQ temp 0.0))
             ((> LOPres 22.0) (SETQ temp 1.0))
             (T (SETQ temp (*$ .1667 (-$ LOPres 16.0)))))
       ; tag 4604
       (SETQ IndicatorV595 (//$ (+$ (*$ IndicatorV595 v)
                                    (*$ temp DeltaT))
                                (+$ v DeltaT)))
       (SETQ OutputAirV595 (+$ (*$ 12.0 IndicatorV595) 3.0))
       (COND (CasualtyUnloadingValveStuckOpen
              (SETQ IndicatorV595 1.0)))))
; tag 4605
```

```
(COND ((AND LOServiceMode
            (NOT CasualtyUnloadingValveStuckOpen))
       (SETQ UnloadingValveFlowCoeff ServiceManualModeLift))
      (T
       (SETQ UnloadingValveFlowCoeff IndicatorV595)))
(SETQ UnloadingValveF//PDropCoeff
      (*$ UnloadingValveCoeff LOTempCoeff UnloadingValveFlowCoeff))
(COND ((> TotalOilHeaderFlow 0.0)
       (COND ((AND (NOT StrainerLockingLugsOnFilter)
                   StrainerLockingLugsOnFilter2
                   (NULL StrainerVentValve2))
              ; ignoring a malop here
              (SETQ FreezeFlag 1)
              (FREEZE 2 1)))))      ; 3 in flowchart
(COND (StrainerFlag
       (COND (StrainerShiftingLever2 (SETQ StrainerFlag NIL)
                                     (SETQ CasualtyCloggedStrainer NIL)
                                     (SETQ ELCLOP NIL))))
      ((NULL StrainerShiftingLever1) (SETQ StrainerFlag T)
                                     (SETQ CasualtyCloggedStrainer NIL)
                                     (SETQ ELCLOP NIL)))    ; at 4610
(COND (CasualtyUnloadingValveStuckShut
       (SETQ ELCDT (+$ ELCDT DeltaT))
       (COND ((AND (> ELCDT 600.0)
                   (OR LOServiceMode
                       (= ServiceManualModeLift 0.0)))
              (SETQ FreezeFlag 2)
              (FREEZE 2 1))))
      (T
       (SETQ ELCDT 0.0)))           ; at 4614
(COND (ELCLOP
       (SETQ StrainerF//PDCoeff 30.0))
      (CasualtyCloggedStrainer
       (SETQ StrainerF//PDCoeff StrainerCasualtyConstant))
      (T
       (SETQ StrainerF//PDCoeff StrainerConstant)))
; at 4623
(SETQ StrainerF//PDCoeff (*$ StrainerF//PDCoeff LOTempCoeff))
(SETQ EmergencyWarmUpF//PDropCoeff
      (*$ EmergencyWarmupConstant LOTempCoeff ELEMS))
(SETQ CoolerF//PDCoeff (*$ CoolerConstant LOTempCoeff))
(COND (CasualtyLOCoolerLeak
       (SETQ CoolerLeakF//PDCoeff (*$ CoolerCasualtyLeakConstant
                                      LOTempCoeff)))
      (T
       (SETQ CoolerLeakF//PDCoeff 0.0)))
(SETQ BearingsF//PDCoeff (*$ TotalBearingF//PConstant LOTempCoeff))
(SETQ a (+$ BearingsF//PDCoeff CoolerLeakF//PDCoeff))
(SETQ b (//$ (*$ CoolerF//PDCoeff a)
             (SQRT (+$ (**$ CoolerF//PDCoeff)
                       (**$ a)))))
(SETQ c (//$ (*$ StrainerF//PDCoeff b)
             (+$ (**$ StrainerF//PDCoeff)
                 (**$ b))))
```

```
(SETQ d (+$ UnloadingValveF//PDropCoeff
           EmergencyWarmupF//PDropCoeff c))
(SETQ e (//$ TotalLOPumpFlow d))
(SETQ StrainerInletPres (**$ e))
(COND ((OR (> StrainerInletPres 62.0)
           (AND (> StrainerInletPres 59.0) ReliefValveOn))
       (SETQ e 7.681)
       (SETQ ReliefValveToSumpFlow (-$ TotalLOPumpFlow (*$ e d)))
       (SETQ StrainerInletPres 59.0)
       (SETQ ReliefValveOn T))
      (T
       (SETQ ReliefValveToSumpFlow 0.0)
       (SETQ ReliefValveOn NIL)))   ; at 4621
(SETQ UnloadingValveToSumpFlow (*$ UnloadingValveF//PDropCoeff e))
(SETQ EmergencyWarmUpFlow (*$ EmergencyWarmupF//PDropCoeff e))
(SETQ TotalOilHeaderFlow (-$ TotalLOPumpFlow
                             UnloadingValveToSumpFlow
                             EmergencyWarmupFlow
                             ReliefValveToSumpFlow))
(SETQ temp (**$ TotalOilHeaderFlow))
(SETQ LOCoolerInletPres (-$ StrainerInletPres
                            (//$ temp (**$ StrainerF//PDCoeff))))
(SETQ ReductionGearFWDHeaderPres
      (MAX 0.0 (-$ LOCoolerInletPres
                   (//$ temp (**$ CoolerF//PDCoeff)))))
(SETQ CoolerCasualtyLostLubeFlow (*$ CoolerLeakF//PDCoeff
                                     (SQRT ReductionGearFWDHeaderPres)))
(SETQ temp (*$ PresRatioLPTurbineInlet ReductionGearFWDHeaderPres))
(SETQ LOPres (//$ (+$ (*$ LOPres p)
                      (*$ temp DeltaT))
                  (+$ p DeltaT)))
(SETQ AttachedPumpSuctionF//PDCoeff (*$ 617.1 LOTempCoeff))
(SETQ AttachedPumpDischargeF//PDCoeff (*$ 184.4 LOTempCoeff))
(SETQ ElectPumpSuctionF//PDCoeff (*$ 385.8 LOTempCoeff))
(SETQ AttachedPUmpSuctionPres
      (-$ (**$ (//$ AttachedLOPumpFlow
                    AttachedPumpSuctionF//PDCoeff))))
(COND ((> AttachedLOPumpFlow 10.0)
       (SETQ AttachedPumpDischargePres
             (+$ StrainerInletPres
                 (**$ (//$ AttachedLOPumpFlow
                           AttachedPumpDischargeF//PDCoeff)))))
      (T
       (SETQ AttachedPumpDischargePres 0.0)))
(for i from 1 to 2
   do (COND ((< (LOPumpFlow i) .0001)
             (store (LOPumpFlow i) 0.0)))
      (store (LOPumpSuctionPres i)
             (-$ (**$ (//$ (LOPumpFlow i)
                           ElectPumpSuctionF//PDCoeff))))
      (store (LOPumpDischargePres i)
             (+$ StrainerInletPres
                 (**$ (//$ (LOPumpFlow i)
                           ElectPumpDischargeF//PDCoeff))))))
```

```
    (COND ((> StrainerInletPres 7.0) (SETQ LOLowPresAlarm NIL))
          (T                         (SETQ LOLowPresAlarm T)))
    (SETQ LOSystemCapacity
          (MAX LowSumpLevel
               (-$ LOSystemCapacity
                   (*$ .01667 CoolerCasualtyLostLubeFlow DeltaT))))
    (COND ((< TotalOilHeaderFlow 25.0)
           (SETQ LOFlowInReductionGear NIL)
           (SETQ LOFlowInLPTurbine NIL)
           (SETQ LOFlowInHPTurbine NIL))
          (T
           (SETQ LOFlowInReductionGear T)
           (SETQ LOFlowInLPTurbine T)
           (SETQ LOFlowInHPTurbine T)))))


(defun LOPresInit ()
    (setq LOServiceMode (select-ic '(nil nil nil nil)))
    (setq ServiceManualModeLift (select-ic '(0.0 0.0 0.0 0.0)))
    (setq LOHeaterValve3 (select-ic '(nil nil nil nil)))
    (setq LOHeaterValve2 (select-ic '(nil nil nil nil)))
    (setq LOHeaterValve1 (select-ic '(nil nil nil nil)))
    (setq StrainerDrainValve3 (select-ic '(nil nil nil nil)))
    (setq StrainerDrainValve2 (select-ic '(nil nil nil nil)))
    (setq StrainerDrainValve1 (select-ic '(nil nil nil nil)))
    (setq StrainerLockingLugsOnFilter (select-ic '(nil nil nil nil)))
    (setq StrainerLockingLugsOnFilter2 (select-ic '(nil nil nil nil)))
    (setq StrainerShiftingLever2 (select-ic '(nil nil nil nil)))
    (setq StrainerShiftingLever1 (select-ic '(t t t t)))
    (setq StrainerJackingScrewsOnCap (select-ic '(nil nil nil nil)))
    (setq StrainerVentValve2 (select-ic '(nil nil nil nil)))
    (setq StrainerVentValve1 (select-ic '(nil nil nil nil)))
    (setq StrainerTrapDoorOpens (select-ic '(nil nil nil nil)))

    (SETQ CasualtyAttachedPumpFail NIL)
    (SETQ CasualtyUnloadingValveStuckOpen NIL)
    (SETQ CasualtyUnloadingValveStuckShut NIL)
    (SETQ CasualtyLOCoolerFailure NIL)
    (SETQ CasualtyLOCoolerLeak NIL)
    (SETQ CasualtyCloggedStrainer NIL)

    (SETQ ELCLOP T)                        ; ELCLOP is a first pass flag
    ; ReliefValve turns on when the pressure is above 62, and off at 59.
    (SETQ ReliefValveOn NIL)

    (SETQ TotalOilHeaderFlow 0.0)
    (SETQ LOServiceUnloadingValve 20.0)
    (SETQ IndicatorV595 0.0)

    ; Model Constants
    (SETQ XMODE2 0)                        ; Accelerated Mode off?
    (SETQ LOServicePumpFlowTC 2.5)
    (SETQ LOUnloadingValveFlowTC 1.5)
    (SETQ LOTurbineInletPresFlowTC 2.0)
```

```
    (SETQ EmergencyWarmupConstant 5.0)
    (SETQ StrainerFlag NIL)
    (SETQ ElectPumpDischargeF//PDCoeff 0.0)
    (SETQ ElectPumpSuctionF//PDCoeff 0.0)
    (SETQ LOPres 0.0)
    (SETQ EmergencyWarmupFlow 0.0)
    (SETQ StrainerF//PDCoeff 0.0)
    (SETQ ELEMS 1.0)                    ; God knows what this is
    (SETQ StrainerCasualtyConstant 130.0)
    (SETQ StrainerConstant 300.0)
    (SETQ CoolerConstant 246.0)
    (SETQ CoolerCasualtyLeakConstant 50.0)
    (SETQ LowSumpLevel 350.0)
    (SETQ AccelModel 16.0)
    (SETQ AccelMode2 16.0)
    (SETQ AccelMode3 16.0)
    (SETQ PresRatioLPTurbineInlet .9648)
    (SETQ TotalBearingF//PConstant 80.269)
    ; Two variables whose initial values may well be dicy...
    (SETQ UnloadingValveCoeff 50.6)
    (SETQ LOSystemCapacity 1200.0)      ; gallons (OK, says the Chief)
    (FILLARRAY LOPumpSuctionPres '(0.0))
    (FILLARRAY LOServicePumpFlowCommand '(0.0))
    (FILLARRAY LOPumpFlow '(0.0))
    (FILLARRAY LOPumpDischargePres '(0.0))
    ())

(defun LOPresStatus ()
    (StatusPrint LOPresStatusVars))

(setq LOPresStatusVars
      '(LOPres IndicatorV595 LOLowPresAlarm TotalOilHeaderFlow
        LOSystemCapacity CasualtyLOCoolerFailure))

(version)
```

B.11 LOPump

```
(version)

(defun LOPumpRun ()
  (PROG ()
    (COND ((NOT LOPumpsPowerOn)
           (for i from 1 to 2
              do (store (LOPumpLowIndicator i) NIL)
                 (store (LOPumpHighIndicator i) NIL)
                 (store (LOPumpOn i) NIL)
                 (store (LOPumpHi//Low i) NIL))
           (GO ServicePurifier)))
    (COND ((EQ Remote//LocalSelect 'Local) (GO ManagePumps))
          ; "IE" is Emergency, "J" is Standby.
```

```
        (Standby//Emergency (SETQ Emergency 1)
                            (SETQ Standby 2))
        (T                  (SETQ Emergency 2)
                            (SETQ Standby 1)))
   (COND ((OR (EQ Manual//AutoSelect 'Manual)
              (> LOPres 14.0))
          (GO ManagePumps)))              ; tag 4410 + 3
   (COND ((OR (> LOPres 13.0)
              (AND (NOT (NCHighStart Emergency))
                   (NOT (> (+$ 1.0 PresCounter3) CounterLimit3))))
          (COND ((> LOPres 13.0) (SETQ PresCounter3 0.0))
                (T (SETQ PresCounter3 (+$ PresCounter3 1.0))
                   (store (LOPumpLowIndicator Emergency) T)
                   (store (NCLowStart Emergency) T))))
        (T
         (COND ((NCHighStart Emergency) (SETQ PresCounter3 0.0))
               (T (SETQ PresCounter3 CounterLimit3)))
         ; tag 7003
         (store (LOPumpLowIndicator Emergency) NIL)
         (store (LOPumpHighIndicator Emergency) T)
         (store (NCLowStart Emergency) NIL)
         (store (NCHighStart Emergency) T)
         (COND ((OR (> LOPres 11.0)
                    (< (+$ PresCounter2 1.0) CounterLimit2))
                (COND ((> LOPres 11.0) (SETQ PresCounter2 0.0))
                      (T (SETQ PresCounter2 (+$ PresCounter2 1.0))))
                (GO ManagePumps)))
         (SETQ PresCounter2 CounterLimit2)
         ; tag 7005 + 4
         (COND ((OR (NOT (< LOPres 10.0))
                    (AND (NCHighStart Standby)
                         (< (+$ 1.0 PresCounter1) CounterLimit1)))
                (COND ((NOT (< LOPres 10.0))
                       (SETQ PresCounter1 0.0))
                      (T
                       (SETQ PresCounter1 (+$ 1.0 PresCounter1))))
                ; tag 7009
                (store (LOPumpLowIndicator Standby) NIL)
                (store (LOPumpHighIndicator Standby) T)
                (store (NCLowStart Standby) NIL)
                (store (NCHighStart Standby) T)))))
ManagePumps
  (for i from 1 to 2
     do (COND ((LOPumpStop i)         ; tag 4415
               (COND ((NCLowStart i)
                      (store (NCLowStart i) NIL)
                      (store (LOPumpLowIndicator i) NIL)))
               (COND ((NCHighStart i)
                      (store (NCHighStart i) NIL)
                      (store (LOPumpHighIndicator i) NIL))))
              ((CasualtyLOPump i)
               (store (LOPumpLowIndicator i) NIL)
               (store (LOPumpHighIndicator i) NIL)
               (store (NCHighStart i) NIL)
```

```
                (store (NCLowStart i) NIL))
               (T                          ; at 4453
                (COND ((LOPumpHighStart i)
                       (COND ((NCLowStart i)
                              (store (NCLowStart i) NIL)
                              (store (LOPumpLowIndicator i) NIL)))
                       (COND ((NOT (NCHighStart i))
                              (store (NCHighStart i) T)
                              (store (LOPumpHighIndicator i) T)))))
                ; Low Start takes priority.
                (COND ((LOPumpLowStart i)
                       (COND ((NCHighStart i)
                              (store (NCHighStart i) NIL)
                              (store (LOPumpHighIndicator i) NIL)))
                       (COND ((NOT (NCLowStart i))
                              (store (NCLowStart i) T)
                              (store (LOPumpLowIndicator i) T)))))))
      ; tag 4454
      (COND ((OR (LOPumpHighIndicator i)
                 (LOPumpLowIndicator i))
             (store (LOPumpOn i) T)
             (COND ((LOPumpLowIndicator i) (store (LOPumpHi//Low i) 'Low))
                   (T                      (store (LOPumpHi//Low i) 'High))))
            (T
             (store (LOPumpOn i) NIL)
             (store (LOPumpHi//Low i) NIL))))
 ServicePurifier
   (COND ((AND LOPurifierPowerOn (NOT LOPurifierStop) LOPurifierStart)
          (SETQ LOPurifierMotorOn T))
         (T
          (SETQ LOPurifierMotorOn NIL)))))


(defun LOPumpInit ()
   (setq LOPurifierStart (select-ic '(nil nil nil nil)))
   (setq LOPurifierStop (select-ic '(nil nil nil nil)))
   (setq Standby//Emergency (select-ic '(nil nil nil nil)))
   (setq Remote//LocalSelect (select-ic '(Remote Remote Remote Remote)))
   (setq Manual//AutoSelect (select-ic '(Manual Manual Auto Auto)))
   (SETQ LOPurifierMotorOn NIL)
   (SETQ LOPurifierPowerOn NIL)
   (SETQ PresCounter1 0.0)
   (SETQ PresCounter2 0.0)
   (SETQ PresCounter3 0.0)
   (SETQ CounterLimit1 2.0)
   (SETQ CounterLimit2 2.0)
   (SETQ CounterLimit3 1.0)
   (FILLARRAY LOPumpHighStart '(()))
   (FILLARRAY LOPumpLowStart '(()))
   (FILLARRAY LOPumpStop '(()))
     ;(setq LOPump2Stop (select-ic '(nil nil nil nil)))
     ;(setq LOPump1Stop (select-ic '(nil nil nil nil)))
     ;(setq LOPump2LowStart (select-ic '(nil nil nil nil)))
     ;(setq LOPump1LowStart (select-ic '(nil nil nil nil)))
```

```
      ;(setq LOPump2HighStart (select-ic '(nil nil nil nil)))
      ;(setq LOPump1HighStart (select-ic '(nil nil nil nil)))
    (FILLARRAY LOPumpLowIndicator '(()))
    (FILLARRAY LOPumpHighIndicator '(()))
    (FILLARRAY LOPumpOn '(()))
    (FILLARRAY LOPumpHi//Low '(()))
    (FILLARRAY NCHighStart '(()))
    (FILLARRAY NCLowStart '(()))
    (FILLARRAY CasualtyLOPump '(()))
    ())

(defun LOPumpStatus ()
    (terpri)
    (princ '|LOPump Status:|)
    (COND ((AND (BOUNDP 'Standby)
                (BOUNDP 'Emergency))
           (format T '|~%~A is Standby; ~A is Emergency|
                     Standby Emergency)))
    (for i from 1 to 2
      do (format T '|~%~A ~:[Off~;Running ~A Speed with flow of ~D gpm~]|
                   i (LOPumpOn i) (LOPumpHi//Low i) (LOPumpFlow i)))
    (format T '|~%Mode is ~A and ~A|
           Manual//AutoSelect Remote//LocalSelect)
    (format T '|~%Purifier motor is ~:[off~;on~]|
           LOPurifierMotorOn))
```

B.12 LOTemp

```
(version)

(declare (unspecial a b c d e) (flonum a b c d e))

(defun LOTempRun ()
  (PROG (a b c d e)
    (COND ((AND LOPurifierMotorOn LOPurifierValve1 LOPurifierValve2
                LOPurifierValve3 LOPurifierValve4 LOPurifierValve9)
           (COND ((AND LOHeaterValve4 LOHeaterValve5)
                  (SETQ Heater//PurifierFlowPathEnabled T)
                  (COND (LOPurifierValve5 (Malop 11 bit23))))
                 (T
                  (SETQ Heater//PurifierFlowPathEnabled NIL)))
           ; tag 3151
           (COND (LOPurifierValve5
                  (SETQ PurifierFlowPathEnabledNoHeater T))
                 (T
                  (SETQ PurifierFlowPathEnabledNoHeater NIL)))
           (COND ((AND (OR LOPurifierValve6
                           LOPurifierValve7
                           LOPurifierValve8)
                       (OR Heater//PurifierFlowPathEnabled
```

```
                        PurifierFlowPathEnabledNoHeater))
              (Malop 11 bit23))))
       (T
        (SETQ Heater//PurifierFlowPathEnabled NIL)
        (SETQ PurifierFlowPathEnabledNoHeater NIL)))
; tag 3110
(COND ((AND (OR CasualtyLOCoolerFailure Heater//PurifierFlowPathEnabled)
            Heater//SteamHeaderPathEnabled)
        (SETQ LOHeaterOn T))
       (T
        (SETQ LOHeaterOn NIL)))       ; Ignoring malop here, at tag 3152
(COND ((OR LOContaminateDrainValve1 LOContaminateDrainValve2
           LOContaminateDrainValve3 LOContaminateDrainValve4)
        (Malop 11 bit12)))
(COND ((OR LOStorageTankValve1 LOStorageTankValve2 LOStorageTankValve3
           LOStorageTankValve4 LOStorageTankValve5)
        (Malop 11 bit12)))
(COND (LOCoolerSWVent (Malop 11 bit0)))
(COND ((AND (OR Heater//PurifierFlowPathEnabled
                PurifierFlowPathEnabledNoHeater)
            (NULL LOPurifierWasteDrain))
        (Malop 11 bit12)))
(SETQ LOHeaterFlow
      (COND (Heater//PurifierFlowPathEnabled
             (+$ EmergencyWarmUpFlow 3.75))
            (T
             EmergencyWarmUpFlow)))
(COND ((= LOHeaterFlow 0.0)
       (SETQ OilIntoHeaterTemp
             (//$ (+$ (*$ LOHeaterTempDecayTC OilIntoHeaterTemp)
                      (*$ RoomTemp DeltaT))
                  (+$ LOHeaterTempDecayTC DeltaT)))
       (SETQ OilOutOfHeaterTemp
             (//$ (+$ (*$ LOHeaterTempDecayTC OilOutOfHeaterTemp)
                      (*$ RoomTemp DeltaT))
                  (+$ LOHeaterTempDecayTC DeltaT))))
      (T
       (SETQ e (MIN 200.0
                    (COND(Heater//SteamHeaderPathEnabled
                          (//$ (*$ 650.0
                                   (-$ HeaterSteamInletValveLift .1))
                               LOHeaterFlow))
                         (T 0.0))))
       (SETQ HeaterTempRise (//$ (+$ (*$ HeaterTempRise LOHeaterTC)
                                     (*$ e DeltaT))
                                 (+$ LOHeaterTC DeltaT)))
       (SETQ OilIntoHeaterTemp MainSumpOilTemp)
       (SETQ OilOutOfHeaterTemp (+$ MainSumpOilTemp HeaterTempRise))))
(SETQ LOCoolerSWFlow
      (COND (LOSWCooler (MIN 1000.0 (*$ MCSWCondFlow
                                        LOCoolerSWToMCRecircFlow
                                        SWCoolerOutletValveLift)))
            (T 0.0)))
(SETQ BearingTempRise (//$ (+$ (*$ BearingTempRise BearingTC)
```

```
                                    (*$ BearingTempFactor
                                        (**$ ShaftRotationRPM)
                                        DeltaT))
                                (+$ BearingTC DeltaT)))
(SETQ OilIntoCoolerTemp (+$ MainSumpOilTemp (*$ .02 TotalOilHeaderFlow)))
(SETQ MainSumpOilTemp
      (-$ (+$ MainSumpOilTemp
              (*$ (+$ (*$ TotalOilHeaderFlow
                          (-$ (+$ BearingTempRise OilOutOfCoolerTemp)
                              OilIntoCoolerTemp))
                      (*$ LOHeaterFlow HeaterTempRise))
                  (//$ DeltaT TempFlowTimeFactor)))
          (*$ (-$ MainSumpOilTemp EngineRoomSeaWaterTemp)
              AmbientHeatLossCoeff
              DeltaT)))
(COND ((> MainSumpOilTemp 260.0) (SETQ MainSumpOilTemp 260.0)))
(COND ((> TotalOilHeaderFlow 1.0)
       ; tag 3154
       (COND ((> LOCoolerSWFlow 1.0)
              ; tag 3156
              (SETQ a (*$ LOCoolerSWFlow TotalOilHeaderFlow))
              (SETQ b (*$ 5.165E-5 a))
              (SETQ c (+$ (*$ .01528 a)
                          (*$ .1327 LOCoolerSWFlow b)))
              (SETQ d (*$ .05758 TotalOilHeaderFlow b))
              (SETQ UnfilteredSWOutletTemp
                    (//$ (+$ (*$ EngineRoomSeaWaterTemp (-$ c d))
                             (*$ OilIntoCoolerTemp (+$ d d)))
                         (+$ c d)))
              (SETQ SWOutletTemp
                    (//$ (+$ (*$ SWOutletTemp LOCoolerTC)
                             (*$ UnfilteredSWOutletTemp DeltaT))
                         (+$ LOCoolerTC DeltaT)))
              (SETQ LOOutletTemp
                    (-$ OilIntoCoolerTemp
                        (//$ (*$ 2.305
                                 LOCoolerSWFlow
                                 (-$ UnfilteredSWOutletTemp
                                     EngineRoomSeaWaterTemp))
                             TotalOilHeaderFlow)))
              (SETQ OilOutOfCoolerTemp
                    (//$ (+$ (*$ OilOutOfCoolerTemp LOCoolerTC)
                             (*$ LOOutletTemp DeltaT))
                         (+$ LOCoolerTC DeltaT))))
             (T
              (SETQ OilOutOfCoolerTemp
                    (//$ (+$ (*$ LOOutletTemp LOCoolerTC)
                             (*$ OilIntoCoolerTemp DeltaT))
                         (+$ LOCoolerTC DeltaT))))))
      ((NOT (> LOCoolerSWFlow 1.0))
       (SETQ OilOutOfCoolerTemp OilIntoCoolerTemp)
       (SETQ SWOutletTemp RoomTemp))
      (T
       (SETQ SWOutletTemp EngineRoomSeaWaterTemp)))
```

```
    ; tag 3155
    (SETQ LOTemp (+$ OilOutOfCoolerTemp BearingTempRise))
    (COND (CasualtyAftJournalBearing
           (SETQ AftJournalBearingTemp
                 (MIN 180.0 (+$ AftJournalBearingTemp DeltaT))))
          (T
           (SETQ AftJournalBearingTemp LOTemp)))
    (COND (CasualtyFwdJournalBearing
           (SETQ FwdJournalBearingTemp
                 (MIN 180.0 (+$ FwdJournalBearingTemp DeltaT))))
          (T
           (SETQ FwdJournalBearingTemp LOTemp)))
    (COND (CasualtyFwdPinionBearing
           (SETQ FwdPinionBearingTemp
                 (MIN 180.0 (+$ FwdPinionBearingTemp DeltaT))))
          (T
           (SETQ FwdPinionBearingTemp LOTemp)))
    (COND (CasualtyMainThrustBearing
           (SETQ MainThrustBearingTemp
                 (MIN 220.0 (+$ MainThrustBearingTemp DeltaT))))
          (T
           (SETQ MainThrustBearingTemp LOTemp)))
    (COND ((OR PurifierFlowPathEnabledNoHeater
               Heater//PurifierFlowPathEnabled)
           (SETQ OilFromHeaterIntoPurifierTemp
                 (//$ (+$ (*$ OilOutOfHeaterTemp
                              3.75
                              (ToFlonum Heater//PurifierFlowPathEnabled))
                          (*$ MainSumpOilTemp
                              3.75
                              (ToFlonum PurifierFlowPathEnabledNoHeater)))
                      (*$ 3.75
                          (+$ (ToFlonum Heater//PurifierFlowPathEnabled)
                              (ToFlonum PurifierFlowPathEnabledNoHeater)))))))
          (T
           (SETQ OilFromHeaterIntoPurifierTemp RoomTemp)))
    (SETQ HeaterTempFlowFactor (+$ .158 (*$ .002526 OilOutOfHeaterTemp)))
    (SETQ PurifierInletPresFactor (*$ PurifierInletPresConstantFactor
                                      HeaterTempFlowFactor))
    (SETQ PurifierInletPres
          (-$ (COND (LOPurifierMotorOn PurifierPresConstant)
                    (T 0.0))
              (//$ (+$ 14.06
                       (COND ((AND Heater//PurifierFlowPathEnabled
                                   (NOT PurifierFlowPathEnabledNoHeater))
                              1.0)
                             (T 0.0)))
                   (**$ PurifierInletPresFactor)))))


(defun LOTempInit ()
    (setq LOSWCooler (select-ic '(nil nil t t)))
    (setq LOHeaterSteamDrain (select-ic '(nil nil t t)))
    (setq LOPurifierWasteDrain (select-ic '(nil t t t)))
```

```
  etq LOCoolerSWVent (select-ic '(nil nil nil nil)))
(setq LOStorageTankValve5 (select-ic '(nil nil nil nil)))
(setq LOStorageTankValve4 (select-ic '(nil nil nil nil)))
(setq LOStorageTankValve3 (select-ic '(nil nil nil nil)))
(setq LOStorageTankValve2 (select-ic '(nil nil nil nil)))
(setq LOStorageTankValve1 (select-ic '(nil nil nil nil)))
(setq LOContaminateDrainValve4 (select-ic '(nil nil nil nil)))
(setq LOContaminateDrainValve3 (select-ic '(nil nil nil nil)))
(setq LOContaminateDrainValve2 (select-ic '(nil nil nil nil)))
(setq LOContaminateDrainValve1 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve9 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve8 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve7 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve6 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve5 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve4 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve3 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve2 (select-ic '(nil nil nil nil)))
(setq LOPurifierValve1 (select-ic '(nil nil nil nil)))
(setq LOHeaterValve5 (select-ic '(nil nil nil nil)))
(setq LOHeaterValve4 (select-ic '(nil nil nil nil)))
(setq SWCoolerOutletValveLift (select-ic '(0.0 0.0 1.0 1.0)))
; HeaterSteamInletValveLift is set in 150psiSteam too.
(setq HeaterSteamInletValveLift (select-ic '(0.0 0.0 0.0 0.0)))

(SETQ CasualtyAftJournalBearing NIL)
(SETQ CasualtyFwdJournalBearing NIL)
(SETQ CasualtyFwdPinionBearing NIL)
(SETQ CasualtyMainThrustBearing NIL)
(SETQ HeaterSteamInletValveLift 0.0)
(SETQ Heater//SteamHeaderPathEnabled NIL)
(SETQ Heater//PurifierFlowPathEnabled NIL)
(SETQ PurifierFlowPathEnabledNoHeater NIL)

(SETQ LOTemp 70.0)
(SETQ AftJournalBearingTemp 70.0)
(SETQ FwdJournalBearingTemp 70.0)
(SETQ FwdPinionBearingTemp 70.0)
(SETQ MainThrustBearingTemp 70.0)
(SETQ OilIntoHeaterTemp 70.0)
(SETQ OilOutOfHeaterTemp 70.0)
(SETQ OilIntoCoolerTemp 70.0)
(SETQ OilOutOfCoolerTemp 70.0)
(SETQ SWOutletTemp 0.0)
(SETQ OilFromHeaterIntoPurifierTemp 70.0)
(SETQ PurifierInletPres 0.0)
(SETQ MainSumpOilTemp 70.0)
(SETQ LOHeaterFlow 0.0)
(SETQ HeaterTempRise 0.0)
(SETQ LOCoolerSWFlow 0.0)
(SETQ BearingTempRise 0.0)
(SETQ UnfilteredSWOutletTemp 70.0)
(SETQ LOOutletTemp 70.0)
(SETQ HeaterTempFlowFactor 70.0)
```

```
    (SETQ PurifierInletPresFactor 0.0)
    (SETQ TimeIncrement .5)
    (SETQ BearingTC 30.0)
    (SETQ BearingTempFactor .00052)
    (SETQ TempFlowTimeFactor 66000.0)
    (SETQ LOCoolerTC 5.0)
    (SETQ RoomTemp 75.0)
    (SETQ LOHeaterTC 10.0)
    (SETQ LOHeaterDecayTC 1200.0)
    (SETQ LOHeaterTempDecayTC 10.0)
    (SETQ PurifierInletPresConstantFactor 2.65)
    (SETQ PurifierPresConstant 25.0)
    (SETQ LOCoolerSWToMCRecircFlow .06)
    (SETQ AmbientHeatLossCoeff 1.0E-5)
    ())


(defun TurnOnLOTemp ()
    (SETQ Heater//SteamHeaderPathEnabled T)
    (SETQ LOPurifierValve1 T)
    (SETQ LOPurifierValve2 T)
    (SETQ LOPurifierValve3 T)
    (SETQ LOPurifierValve4 T)
    (SETQ LOPurifierValve9 T) ; Turn on Heater//PurifierFlowPath
    (SETQ LOHeaterValve4 T)
    (SETQ LOHeaterValve5 T)   ; Turn on the sea water to the LOCooler
    ())


(defun LOTempStatus ()
    (StatusPrint LOTempStatusVars))

(setq LOTempStatusVars '(OilIntoHeaterTemp OilOutOfHeaterTemp
                         OilIntoCoolerTemp OilOutOfCoolerTemp
                         MainSumpOilTemp LOOutletTemp PurifierInletPres))
```

B.13 MainCondenser

```
(version)

(declare (unspecial sumq d temp) (flonum sumq d temp))

(defun MainCondenserRun ()
 (PROG (sumq d temp)
     (SETQ HotwellLiquidVolume (*$ .01625 HotwellLiquidMass))
     (SETQ HotwellWaterLevel (*$ .4646 HotwellLiquidVolume))
     (COND ((< HotwellWaterLevel 0.0) (SETQ HotwellWaterLevel 0.0)))
     (COND ((< HotwellLiquidVolume 19.78) (SETQ SeaWaterFlowCoeff 1.0))
           (T (SETQ HotwellWaterLevel
                    (+$ (*$ .1498 HotwellLiquidVolume) 6.226))
              (COND ((< HotwellWaterLevel 17.0)
```

```
                (SETQ SeaWaterFlowCoeff 1.0))
               (T
                (COND ((> HotwellWaterLevel 104.0)
                       (SETQ HotwellWaterLevel 104.0)))
                (SETQ SeaWaterFlowCoeff
                      (FBOUNDS 0.0
                               (+$ (*$ HotwellWaterLevel -.2327)
                                   5.502)
                               1.546))))))
; tag 3251
(SETQ HotwellLevelGage (-$ HotwellWaterLevel 3.0))

; Calculate steam mixture enthalpy.
(SETQ TotalSteamFlowInput (+$ ForwardSteamFlow
                              MainCondenserSteamFlow
                              AsternSteamFlow))
(COND ((< TotalSteamFlowInput .01)        ; Cold Iron test
       (SETQ TotalSteamFlowInput .01)))
(COND ((AND (= MainShaftNormalizedRPM 0.0)
            ; ShaftRotation is a guess, the original listing
            ; has ETTNMA, which has not been found yet.
            (= CircPumpMotorSpeed 0.0)
            (= ShipSpeed 0.0)
            (= CondensateFlowAEEnthalpy 0.0))
       (SETQ temp 1.0))
      ((AND (< MCSWCondFlow CoolingFlowThreshold)
            (= CircPumpMotorSpeed 0.0))
       (SETQ temp
             (FBOUNDS 0.0
                      (//$ (-$ MCSWCondFlow
                               CoolingFlowIntercept)
                           (-$ CoolingFlowThreshold
                               CoolingFlowIntercept))
                      1.0))
       (SETQ CondensateFlowAEEnthalpy OverheatConstant))
      (T
       (SETQ temp 1.0)
       (SETQ CondensateFlowAEEnthalpy 0.0)))

    ;; Now at tag 3254, and assuming SUMQ is an internal variable.
    (SETQ sumq
          (+$ (*$ ForwardSteamFlow TurbineExitEnthalpy)
              (*$ MainCondenserSteamFlow MainCondenserSteamEnthalpy)
              (*$ CondensateFlowAECondenser CondensateFlowAEEnthalpy)
              (*$ AsternSteamFlow TurbineExhaustEnthalpy)))
    (SETQ SteamAverageEnthalpy (//$ sumq TotalSteamFlowInput))
    (SETQ FilteredHeatExchangerValue SteamAverageEnthalpy)
    (SETQ EffectiveSeaWaterFlow
          (*$ temp SeaWaterFlowCoeff SeaWaterCoolantMassFlow))
    (COND ((< EffectiveSeaWaterFlow 1.0)
           (SETQ EffectiveSeaWaterFlow 1.0)))
    (COND ((< SeaWaterCoolantMassFlow 1.0)
           (SETQ SeaWaterCoolantMassFlow 1.0)))
    (SETQ CondenserHeatTransferCoeff
```

```
       *$ 20.45 (SQRT SeaWaterCoolantMassFlow)))

; Calculate steam temperature.
(SETQ HeatExchangerFunction0
      (//$ 1.0 (+$ (//$ 1.0 CondenserHeatTransferCoeff)
                   (//$ 1.0 (*$ 2.0 EffectiveSeaWaterFlow)))))
(SETQ CondenserLiquidTC (//$ SeaWaterMass
                             (*$ DeltaT SeaWaterCoolantMassFlow)))
(SETQ HeatExchangerFunction1 (//$ HeatExchangerFunction0
                                  TotalSteamFlowInput))
(SETQ HeatExchangerFunction2 (//$ 1.0 HeatExchangerFunction1))
(SETQ UnfilteredHotwellTemp
      (FBOUNDS EngineRoomSeaWaterTemp
               (+$ (//$ (+$ FilteredHeatExchangerValue 32.0)
                        (+$ HeatExchangerFunction1 1.0))
                   (//$ EngineRoomSeaWaterTemp
                        (+$ HeatExchangerFunction2 1.0)))
               300.0))
(SETQ HotwellTemp (//$ (+$ (*$ TempTC1 HotwellTemp)
                           (*$ UnfilteredHotwellTemp DeltaT))
                       (+$ TempTC1 DeltaT)))

; Calculate sea water exit temperature.
(SETQ HeatExchangerFunction3 (//$ (*$ 2.0 EffectiveSeaWaterFlow)
                                  CondenserHeatTransferCoeff))
(SETQ SeaWaterOutletTemp
      (*$ (//$ 1.0 (+$ 1.0 HeatExchangerFunction3))
          (-$ (*$ 2.0 HotwellTemp)
              (*$ (-$ 1.0 HeatExchangerFunction3)
                  EngineRoomSeaWaterTemp))))
(COND ((> SeaWaterOutletTemp 220.0)
       (SETQ SeaWaterOutletTemp 220.0)))
(COND (FirstPassFlag           ; Initially T
       (SETQ SeaWaterCoolantOutletTemp SeaWaterOutletTemp))
      (T
       (SETQ SeaWaterCoolantOutletTemp
             (*$ (//$ 1.0 (+$ 1.0 CondenserLiquidTC))
                 (+$ SeaWaterOutletTemp
                     (*$ CondenserLiquidTC
                         SeaWaterCoolantOutletTemp)))))))
; tag 3210

;; Assuming ECTVD is a temporary, using d instead.
(SETQ d (+$ CombinedTempInterceptConstant
            (*$ CombinedTempSlopeConstant
                (+$ SteamFlowRatioForward
                    (*$ SteamFlowRatioAstern .00046)))))
(COND ((< d UnfilteredHotwellTemp)
       (SETQ UnfilteredVaporTemp UnfilteredHotwellTemp))
      (T
       (SETQ UnfilteredVaporTemp d)))
(SETQ LPTurbineExhaustTempCondenser
      (//$ (+$ (*$ LPTurbineExhaustTempCondenser TempTC2)
               (*$ DeltaT UnfilteredVaporTemp))
```

```
                *$ TempTC2 DeltaT)))
        (COND ((> LPTurbineExhaustTempCondenser 212.0)
               (SETQ LPTurbineExhaustTempCondenser 212.0)))

        ;; The Previous expression was before the conditional on the
        ;; CondensateFlowAEEnthalpy, which seemed to cause
        ;; problems.

        (COND ((> HotwellTemp LPTurbineExhaustTempCondenser)
               (SETQ HotwellTemp LPTurbineExhaustTempCondenser)))
        (SETQ CondenserSteamPres
              (-$ (*$ LPTurbineExhaustTempCondenser
                      (+$ .1361
                          (*$ -.001392 LPTurbineExhaustTempCondenser)
                          (*$ 5.5E-6
                              (**$ LPTurbineExhaustTempCondenser))))
                  4.34))
        (SETQ LPTurbineExhaustPresCondenser
              (+$ CondenserAirPres (*$ CondenserSteamPres 2.0354)))
        (COND ((AND (> TotalSteamFlowInput 1.0)
                    (> LPTurbineExhaustPresCondenser VacuumThreshold)
                    FirstPassFlag)     ; Why test FirstPassFlag?
               (SETQ FreezeFlag 5)
               (FREEZE 2 1)))
        (SETQ VacuumVaporPres
              (-$ 29.92 LPTurbineExhaustPresCondenser))
        (SETQ HotwellBottomPresHead
              (+$ (*$ LPTurbineExhaustPresCondenser .4913)
                  (*$ HotwellWaterLevel .03588)
                  -14.7))
        (SETQ WaterMassDerivative
              (+$ RecircFlow
                  TotalSteamFlowInput
                  (*$ (-$ CondenserFillFlow CondensateFlow) .13369)))
        (SETQ HotwellLiquidMass
              (+$ HotwellLiquidMass (*$ WaterMassDerivative DeltaT)))
        (COND ((< HotwellLiquidMass 1.0) (SETQ HotwellLiquidMass 1.0)))
        (SETQ FirstPassFlag NIL)
        ))


(defun MainCondenserInit ()
    (SETQ CondenserFillFlow 0.0)
    (SETQ FilteredHeatExchangerValue 0.0)
    (SETQ EngineRoomSeaWaterTemp 60.0)  ; says The Chief
    (SETQ MainCondenserSteamEnthalpy 0.0)
    (SETQ 150psiSteamEnthalpy 1200.0)   ; from FireMain Integration
    (SETQ FirstPassFlag T)
    (SETQ HotwellLiquidVolume 0.0)
    (SETQ HotwellWaterLevel 0.0)
    (SETQ HotwellLiquidMass 0.0)
    (SETQ SeaWaterFlowCoeff 0.0)
    (SETQ TotalSteamFlowInput 0.0)
    (SETQ CondensateFlowAEEnthalpy 0.0)
```

```
    (SETQ SteamAverageEnthalpy 0.0)
    (SETQ FilteredHeatExchangerValue 0.0)
    (SETQ EffectiveSeaWaterFlow 0.0)
    (SETQ SeaWaterCoolantMassFlow 0.0)
    (SETQ CondenserHeatTransferCoeff 0.0)
    (SETQ HeatExchangerFunction0 0.0)
    (SETQ HeatExchangerFunction1 0.0)
    (SETQ HeatExchangerFunction2 0.0)
    (SETQ HeatExchangerFunction3 0.0)
    (SETQ CondenserLiquidTC 0.0)
    (SETQ UnfilteredHotwellTemp 0.0)
    (SETQ HotwellTemp 70.0)
    (SETQ SeaWaterOutletTemp 0.0)
    (SETQ CondenserSteamPres 0.0)
    (SETQ LPTurbineExhaustTempCondenser 0.0)
    (SETQ LPTurbineExhaustPresCondenser 0.0)
    (SETQ VacuumVaporPres 0.0)
    (SETQ HotwellBottomPresHead 0.0)
    (SETQ WaterMassDerivative 0.0)

    ;; Model Constants
    (SETQ SeaWaterMass 12500.0)
    (SETQ TempTC1 30.0)
    (SETQ TempTC2 50.0)
    (SETQ DischargeTempCurveConstant .9)
    (SETQ DischargeTempCurveSlopeConstant .5)
    (SETQ DischargeTempInterceptConstant 83.0)
    (SETQ CombinedTempSlopeConstant .45)
    (SETQ CombinedTempInterceptConstant 74.7)
    (SETQ VacuumThreshold 20.0)
    (SETQ CoolingFlowThreshold 12000.0)
    (SETQ OverheatConstant 1000.0)
    (SETQ CoolingFlowIntercept 4000.0)
    (SETQ HiRecircHeatLossConstant .5)
    ())

(defun MainCondenserStatus ()
    (StatusPrint MainCondenserStatusVars))

(defun MainCondenserValveStatus ()
    (StatusPrint MainCondenserValveStatusVars))

(setq MainCondenserParameters '(RecircFlowDemand
                                CoolingWaterPresHead
                                CondensateFlowAECondenser
                                RecircFlow))

(setq MainCondenserStatusVars '(HotwellWaterLevel
                                HotwellTemp
                                LPTurbineExhaustTempCondenser
                                LPTurbineExhaustPresCondenser
                                CondenserSteamPres
                                UnfilteredHotwellTemp
                                UnfilteredVaporTemp
```

```
                                CondenserAirPres
                                CondensateFlow
                                SeaWaterCoolantOutletTemp))

(setq MainCondenserValveStatusVars
      '(StartCondPump1A       StartCondPump1B
        CondPump1APowerOn     CondPump1BPowerOn
        CondPump1AAuralCue    CondPump1BAuralCue
        Pump1ASuctionValve    Pump1BSuctionValve
        Pump1AGlandSealValve  Pump1BGlandSealValve
        Pump1AVentValve       Pump1BVentValve
        Pump1ADischargePres   Pump1BDischargePres
        CondensateSystemShutoffValve))
```

B.14 MainSteam

```
(version)

(declare (unspecial a b c d e j) (flonum a b c d e j))

(defun MainSteamRun ()
    (PROG (a b c d e j)
      (SETQ MSSteamDemand
            (+$ (*$ .603
                    SteamFlowRatioForward
                    (COND ((EQ AheadEnable 'SECURED) 0.0)
                          ((EQ AheadEnable 'OPERATING) 1.0)
                          (T 2.0)))
                (*$ .00028
                    SteamFlowRatioAstern
                    (COND ((EQ AsternEnable 'SECURED) 0.0)
                          ((EQ AsternEnable 'OPERATING) 1.0)
                          (T 2.0)))))
      ; MTAheadSteam is in terms of lbs/sec.
      (COND ((EQ Mode 'Integrated)
             ; Won't happen for a while...
             ;  MSCrossConnect=EMVCC
             ;  MSSteamEnable=EME
             ;  TurbineInletSteamPres=ETTIP
             ;  TurbineInletSteamTemp=ETTIT
             ;  TurbineHeaderPres=EMPMS

             (SETQ MSCrossConnect FRCrossConnect)
             (SETQ MSSteamEnable FRSteamEnable)
             (SETQ TurbineInletSteamPres FRMainSteamPres)
             (SETQ TurbineInletSteamTemp FRMainSteamTemp)
             (SETQ TurbineHeaderPres FRMainSteamPres)
             (SETQ CrossConnect1 StopValve1MS2A)
             (SETQ CrossConnect2 StopValve1MS2B)
             (SETQ CrossConnect3 (OR StopValve1MS2A WarmupBypass1MS2A))
             (SETQ CrossConnect4 (OR StopValve1MS2B WarmupBypass1MS2B)))
```

```
(T                          ; INDEPENDENT mode
 (SETQ MSCrossConnect T)
 (SETQ SteamPressureLevel (*$ 1300.0 SteamPressureKnob))
 ; Instructor's controls we call "knobs".
 (SETQ 150lbSteamReducerPres (-$ SteamPressureLevel 10.0))
 (COND ((NOT (OR StopValvelMS2A StopValvelMS2B
                 WarmupBypasslMS2A WarmupBypasslMS2B))
        ; Engine room secured
        (SETQ MSSteamEnable 'SECURED)
        (SETQ TurbineHeaderPres 0.0)
        (SETQ TurbineInletSteamTemp 70.0)
        (SETQ TurbineInletSteamPres 0.0))
       (T
        (COND ((OR StopValvelMS2A StopValvelMS2B)
               (SETQ MSSteamEnable 'OPERATING))
              (T
               (SETQ MSSteamEnable 'WARMUP)))
        (SETQ TurbineInletSteamTemp 950.0)
        (COND (ControlPanelACCOn
               ; steam pressure rises.
               (SETQ b (*$ DeltaT (+$ EMT1 EMT2)))
               (SETQ c (*$ DeltaT DeltaT))
               (SETQ d (*$ EMT3 DeltaT (-$ MSSteamDemand
                                           IMPresTrans4)))
               (SETQ d (//$ (+$ (*$ IMPresTrans2
                                    (+$ (*$ 2.0 EMT6) b))
                                (-$ (*$ EMT6 IMPresTrans3)) d)
                            (+$ EMT6 b c)))
               (SETQ IMPresTrans4 MSSteamDemand)
               (SETQ IMPresTrans3 IMPresTrans2)
               (SETQ IMPresTrans2 d)
               (SETQ IMPresTransl 0.0)
               (COND ((< d 0.0)
                      (SETQ d 0.0)
                      (SETQ IMPresTrans2 0.0)
                      (SETQ IMPresTrans3 0.0)
                      (SETQ IMPresTrans4 0.0)))
               (SETQ TurbineHeaderPres (-$ 1265.0 d))
               (SETQ TurbineInletSteamPres 1265.0))
              (T            ; ControlPanelAccOn is NIL
               (SETQ IMPresTransl
                     (//$ (+$ (*$ EMT4 IMPresTransl)
                              (*$ EMKW MSSteamDemand
                                  DeltaT))
                          (+$ EMT4 DeltaT)))
               (SETQ TurbineHeaderPres
                     (FBOUNDS 0.0
                              (-$ SteamPressureLevel
                                  IMPresTransl)
                              1300.0))
               (SETQ IMPresTransl (-$ SteamPressureLevel
                                      TurbineHeaderPres))
               (SETQ IMPresTrans4 MSSteamDemand)
               (SETQ IMPresTrans3 0.0)
```

```
                    (SETQ IMPresTrans2 0.0)
                    (SETQ TurbineInletSteamPres
                          SteamPressureLevel)))))))

;; Now at 3 in flowchart, tag 4110 in code.
(COND ((NOT (OR MSWarmupDrain1 MSWarmupDrain2))
       (COND (MSConstantDrain (SETQ MSDrainAlign 'OPERATING))
             (T               (SETQ MSDrainAlign 'SECURED))))
      ((OR (XOR MSWarmupDrain1 MSWarmupDrain2)
           MSConstantDrain)
       (Malop emalop 7 bit12))
      (T
       (SETQ MSDrainAlign 'WARMUP)))
(COND ((NEQ MSSteamEnable MSDrainAlign)
       (Malop emalop 7 bit12)))
(COND ((OR (NOT (AND (EQ MSSteamEnable 'OPERATING)
                     StopValve1MS2A
                     StopValve1MS2B))
           StopValve1MS2A
           StopValve1MS2B)
       (Malop emalop 7 bit03)))
(COND (MSCrossConnect (SETQ c 1.17)
                      (SETQ d .0943))
      (T (SETQ c .771)
         (SETQ d .0489)))
(SETQ TurbineHeaderPres
      (-$ TurbineHeaderPres (*$ MSSteamDemand
                                (+$ c (*$ MSSteamDemand d)))))
(COND ((< TurbineHeaderPres 0.0) (SETQ TurbineHeaderPres 0.0)))
(COND ((EQ MSSteamEnable 'SECURED)          (SETQ a 1000.0))
      ((EQ MSSteamEnable 'OPERATING)        (SETQ a 1.0))
      ; assume WARMUP
      ((AND MSWarmupDrain1 MSWarmupDrain2) (SETQ a 15.0))
      (T                                    (SETQ a 1000.0)))
(SETQ TurbineHeaderTemp (//$ (+$ (*$ a  TurbineHeaderTemp)
                                 (*$ DeltaT
                                     TurbineInletSteamTemp))
                             (+$ a DeltaT)))
; now at line 1427
(COND (AheadGuarding              (SETQ AheadEnable 'OPERATING))
      (AheadGuardingWarmupBypass (SETQ AheadEnable 'WARMUP))
      (T                         (SETQ AheadEnable 'SECURED)))
(COND ((EQ MSSteamEnable 'SECURED)
       (SETQ AheadEnable 'SECURED))
      ((EQ MSSteamEnable 'WARMUP)
       (COND ((NEQ AheadEnable 'SECURED)
              (SETQ AheadEnable 'WARMUP)))))
(COND (AsternGuarding              (SETQ AsternEnable 'OPERATING))
      (AsternGuardingWarmupBypass (SETQ AsternEnable 'WARMUP))
      (T                          (SETQ AsternEnable 'SECURED)))
(COND ((EQ MSSteamEnable 'SECURED)
       (SETQ AsternEnable 'SECURED))
      ((EQ MSSteamEnable 'WARMUP)
       (COND ((NEQ AsternEnable 'SECURED)
```

```
                    (SETQ AsternEnable 'WARMUP)))))
      ; line 1438
      (COND ((NOT (OR AheadGuardingWarmupBypass AsternGuardingWarmupBypass
                      SteamChestWarmupBypass1 SteamChestWarmupBypass2))
             (COND (SteamChestContinuousDrain
                    (AND (NEQ AheadEnable 'OPERATING)
                         (Malop 13 Drain Engine)))
                   ((NEQ AheadEnable 'SECURED)
                    (Malop 13 Drain Engine))))
            ((AND AheadGuardingWarmupBypass AsternGuardingWarmupBypass
                  SteamChestWarmupBypass1 SteamChestWarmupBypass2
                  (NOT SteamChestContinuousDrain))
             (COND ((NEQ AheadEnable 'WARMUP)
                    (Malop 13 Drain Engine))))
            (T
             (Malop Emalop 7 ibit12)))
      (COND ((NOR (AND (NOR MSCasingDrain1 MSCasingDrain2)
                       (OR (EQ AheadEnable 'SECURED)
                           (> MSSteamDemand 0.0)))
                  (AND MSCasingDrain1 MSCasingDrain2
                       (= MSSteamDemand 0.0)
                       (NEQ AheadEnable 'OPERATING)))
             (Malop 7 Bit12)))
      (SETQ j (AND (> MSValveStrokeForward .143)
                   (NEQ AheadEnable 'SECURED)))
      (COND ((EQ MSSteamEnable 'SECURED)
             (AND (NOT (OR MSWarmupDrain1 MSWarmupDrain2
                           (AND (OR AheadGuarding AsternGuarding)
                                (OR SteamChestWarmupBypass1
                                    SteamChestWarmupBypass2))
                           j))
                  (SETQ MSHeaderPres (//$ (*$ MSHeaderPres
                                              MSDecaySteamPressure)
                                          (+$ MSHeaderPres DeltaT)))))
            ((EQ MSSteamEnable 'OPERATING)
             (SETQ MSHeaderPres TurbineHeaderPres))
            (T          ; WARMUP
             (SETQ MSHeaderPres
                   (COND (j (//$ (*$ TurbineHeaderPres DeltaT)
                                 (+$ EMT5 DeltaT)))
                         (T (//$ (+$ (*$ TurbineHeaderPres DeltaT)
                                     (*$ MSHeaderPres EMT5))
                                 (+$ EMT5 DeltaT))))))))
      (SETQ MSSteamDemandRTI MSSteamDemand)))


(defun MainSteamInit ()
   (setq SteamPresKnob (select-ic '(0.0 0.0 1.0 1.0)))
   (setq ControlPanelACCOn (select-ic '(nil t t t)))
   (setq MSCasingDrain2 (select-ic '(nil nil nil nil)))
   (setq MSCasingDrain1 (select-ic '(nil nil nil nil)))
   (setq SteamChestContinuousDrain (select-ic '(nil nil t t)))
   (setq SteamChestWarmupBypass2 (select-ic '(nil nil nil nil)))
   (setq SteamChestWarmupBypass1 (select-ic '(nil nil nil nil)))
```

```
    (setq AsternGuardingWarmupBypass (select-ic '(nil nil nil nil)))
    (setq AheadGuardingWarmupBypass (select-ic '(nil nil nil nil)))
    (setq AsternGuarding (select-ic '(nil nil t t)))    ; also in MainTurbine
    (setq AheadGuarding (select-ic '(nil nil t t)))
    ; AheadGuarding is also in MainTurbine and Controlair
    (setq MSValveStrokeForward
          (select-ic '(0.0 0.0 0.276 0.57)))    ; also in MainTurbine
    (setq MSConstantDrain (select-ic '(nil nil t t)))
    (setq MSWarmupDrain2 (select-ic '(nil nil nil nil)))
    (setq MSWarmupDrain1 (select-ic '(nil nil nil nil)))
    (setq WarmupBypass1MS2B (select-ic '(nil nil nil nil)))
    (setq WarmupBypass1MS2A (select-ic '(nil nil nil nil)))
    (setq StopValve1MS2B (select-ic '(nil nil t t)))
    (setq StopValve1MS2A (select-ic '(nil nil t t)))

    (SETQ mode 'INDEPENDENT)
    (SETQ EMT1 3.0)
    (SETQ EMT2 10.0)
    (SETQ EMT3 50.0)
    (SETQ EMT4 10.0)
    (SETQ EMT5 5.0)
    (SETQ EMT6 (*$ EMT1 EMT2))
    (SETQ EMKW 8.0)
    (SETQ MSDecaySteamPressure 20.0)
    (SETQ SteamPressureLevel 0.0)
    (SETQ 150lbSteamReducerPres 0.0)
    (SETQ TurbineInletSteamPres 0.0)
    (SETQ TurbineInletSteamTemp 0.0)
    (SETQ MTAheadSteam 0.0)
    (SETQ MTAsternSteam 0.0)
    (SETQ DeltaT 1.0)
    (SETQ AheadEnable 'SECURED)
    (SETQ AsternEnable 'SECURED)
    (SETQ MSSteamEnable 'SECURED)
    (SETQ TurbineHeaderTemp 700.0)      ; Chief Roe
    (SETQ TurbineHeaderPres 1150.0)     ; Chief Roe
    (SETQ MSHeaderPres 1200.0)          ; Chief Roe
    (SETQ IMPresTrans1 0.0)
    (SETQ IMPresTrans2 0.0)
    (SETQ IMPresTrans3 0.0)
    (SETQ IMPresTrans4 0.0)
    ())

(defun MainSteamWarmup ()
    (SETQ WarmupBypass1MS2A T)
    (SETQ WarmupBypass1MS2B T)
    (SETQ AheadGuardingWarmupBypass T)
    (SETQ AsternGuardingWarmupBypass T)
    (SETQ SteamChestWarmupBypass1 T)
    (SETQ SteamChestWarmupBypass2 T)
    (SETQ MSWarmupDrain1 T)
    (SETQ MSWarmupDrain2 T)
    ())
```

```
(defun MainSteamOperate ()
   (SETQ WarmupBypass1MS2A NIL)
   (SETQ WarmupBypass1MS2B NIL)
   (SETQ StopValve1MS2A T)
   (SETQ StopValve1MS2B T)
   (SETQ SteamChestWarmupBypass1 NIL)
   (SETQ SteamChestWarmupBypass2 NIL)
   (SETQ MSWarmupDrain1 NIL)
   (SETQ MSWarmupDrain2 NIL)
   (SETQ AheadGuardingWarmupBypass NIL)
   (SETQ AsternGuardingWarmupBypass NIL)
   (SETQ AheadGuarding T)
   (SETQ AsternGuarding T)
   (SETQ MSConstantDrain T)
   ; Chief Ski claims these last three valves remain open up
   ; to 1/3 Ahead throttle, and for Astern operation.
   (SETQ MSCasingDrain1 T)
   (SETQ MSCasingDrain2 T)
   (SETQ SteamChestContinuousDrain T)
   ())

(defun MainSteamStatus ()
   (StatusPrint MainSteamStatusVars))

(setq MainSteamStatusVars
      '(ShipSpeedKnots ShaftRotationRPM
        AheadGuarding AsternGuarding
        TurbineHeaderPres TurbineHeaderTemp
        TurbineInletSteamPres TurbineInletSteamTemp
        AheadEnable AsternEnable))
```

B.15 MainTurbine

```
(declare (unspecial a b c temp) (flonum a b c temp))

(defun MainTurbineRun ()
 (PROG (a b c temp)
     (COND ((NOT CasualtyMainThrottleInoperative)
            (COND ((> MSValveStrokeForward .143)
                   (SETQ ForwardThrottle T))
                  (T
                   (SETQ ForwardThrottle NIL)))
            (SETQ MainThrottleForwardValveLift
                  (*$ 1.282 (-$ MSValveStrokeForward .143)))
            (COND ((< MainThrottleForwardValveLift 0.0)
                   (SETQ MainThrottleForwardValveLift 0.0))
                  ((> MainThrottleForwardValveLift 1.0)
                   (SETQ MainThrottleForwardValveLift 1.0)))
            (SETQ StrokeCrossbarMainThrottleForward
                  MainThrottleForwardValveLift)))
```

```
(COND ((EQ AheadEnable 'OPERATING)
       (SETQ SteamFlowRatioForward
             (*$ 100.0 MainThrottleForwardValveLift))
       (COND ((< SteamFlowRatioForward 50.0)
              (SETQ a -.00011764)
              (SETQ b (+$ 1.152 (*$ SteamFlowRatioForward -.0002))))
             (T
              (SETQ a (+$ -.000174056
                          (*$ SteamFlowRatioForward 1.128E-6)))
              (SETQ b (+$ 1.212 (*$ SteamFlowRatioForward -.001552))))))
      (T
       (SETQ SteamFlowRatioForward 0.0)
       (SETQ a -.00011764)
       (SETQ b (+$ 1.152 (*$ SteamFlowRatioForward -.0002)))))
; tag 3052
(SETQ FB (+$ (*$ a TurbineHeaderPres) b))
(SETQ FD (+$ .91 (*$ LPTurbineExhaustPresCondenser .4912
                     (+$ .00044 (//$ 1.5148 (+$ SteamFlowRatioForward
                                                4.04))))))
(SETQ SupplySteamTempFunction
      (+$ 1.0 (*$ SupplySteamTransientTemp
                  (-$ TurbineHeaderTemp 950.0))))
(COND ((< TurbineHeaderPres 1000.0)
       (SETQ SupplySteamPresFunction
             (-$ 1.0 (*$ -1.0E-6 (-$ 1000.0 TurbineHeaderPres)))))
      (T
       (SETQ SupplySteamPresFunction 1.0)))
(SETQ SteamFlowModified (//$ (*$ SteamFlowRatioForward
                                 SupplySteamTempFunction
                                 SupplySteamPresFunction)
                             (*$ FD FB)))
(COND ((AND ForwardThrottle (EQ AheadEnable 'OPERATING)
            (OR (< TurbineHeaderPres MinValveSupplySteamPres)
                (< TurbineHeaderTemp MinValveSupplySteamTemp)))
       (SETQ FreezeFlag 4)
       (FREEZE 2 1)))
(SETQ LPTurbineExhaustPres
      (-$ -1.188 (//$ 347.513 (-$ SteamFlowRatioForward 195.52))))
(SETQ TurbineDeltaPresExit
      (-$ LPTurbineExhaustPres
          (*$ LPTurbineExhaustPresCondenser .4912)))
(SETQ a (-$ 10.5 (//$ 1983.0 (+$ 183.0 SteamFlowRatioForward))))
(SETQ b (+$ -31.62 (//$ 10667.0 (+$ 98.94 SteamFlowRatioForward))))
(COND ((< TurbineDeltaPresExit 0.0)
       (SETQ temp (-$ TurbineDeltaPresExit)))
      (T
       (SETQ temp TurbineDeltaPresExit)))
(SETQ TurbineDeltaEnthalpyExit
      (+$ (*$ a TurbineDeltaPresExit) (*$ b (SQRT temp))))
(SETQ LPTurbineExhaustEnthalpy
      (+$ 1100.0
          (*$ -3.056 SteamFlowRatioForward)
          (*$ .02504 (**$ SteamFlowRatioForward))))
(SETQ TurbineExitEnthalpy
```

```
      (-$ LPTurbineExhaustEnthalpy TurbineDeltaEnthalpyExit))
(SETQ ForwardSteamFlow (*$ .6028 SteamFlowModified))
(COND (AheadGuarding
       (SETQ AheadSteamChestPres (*$ .97 TurbineHeaderPres))
       (SETQ HPTurbineExhaustPres (-$ SteamFlowModified 20.0))
       (SETQ StagelHPPres (-$ (*$ 6.51 SteamFlowModified) 15.0))
       (COND (ForwardThrottle
              (SETQ StagelHPTemp 400.0))
             (T
              (SETQ StagelHPTemp (+$ (*$ 3.0 SteamFlowModified) 553.0))))
       (SETQ HPTurbineExhaustTemp
             (*$ StagelHPTemp .34 TurbineHeaderPres)))
      (T                              ; Ahead Guarding off
       (SETQ AheadSteamChestPres 0.0)
       (SETQ HPTurbineExhaustPres 0.0)
       (SETQ StagelHPPres 0.0)
       (COND (ForwardThrottle (SETQ StagelHPTemp 400.0))
             (T                (SETQ StagelHPTemp 0.0)))
       (SETQ HPTurbineExhaustTemp 0.0)))
(COND ((> HPTurbineExhaustPres 70.0) (SETQ HPTurbineExhaustPres 70.0))
      ((< HPTurbineExhaustPres 0.0)  (SETQ HPTurbineExhaustPres 0.0)))
(COND ((< SteamFlowModified 70.0)
       (SETQ a (*$ 2.6 SteamFlowModified))
       (SETQ b (*$ -.4 SteamFlowModified)))
      (T
       (SETQ a (-$ (*$ 5.5 SteamFlowModified) 200.0))
       (COND ((< SteamFlowModified 75.0)
              (SETQ b (*$ SteamFlowModified -.4)))
             (T
              (SETQ b (+$ (*$ -1.3 SteamFlowModified) 67.5))))))
; tag 3005
(SETQ ShaftTorqueForward (+$ a (*$ b ShaftRotation)))
(COND ((> MSValveStroke .133) (SETQ AsternThrottle T))
      (T                      (SETQ AsternThrottle NIL)))
(SETQ MainThrottleAsternValveLift
      (FBOUNDS 0.0 (*$ 1.613 (-$ MSValveStroke .133)) 1.0))
(COND ((EQ AsternEnable 'OPERATING)
       (SETQ AsternSteamFlowFunction
             (*$ 100000.0 (**$ MainThrottleAsternValveLift))))
      (T
       (SETQ AsternSteamFlowFunction 0.0)))
(SETQ SteamFlowRatioAstern (//$ (*$ AsternSteamFlowFunction
                                    SupplySteamTempFunction
                                    SupplySteamPresFunction
                                    SteamFlowConstantReverse)
                                (*$ FB FD)))
(COND (AsternGuarding (SETQ LPTurbineInletPresAstern
                            (*$ SteamFlowRatioAstern .007)))
      (T              (SETQ LPTurbineInletPresAstern 0.0)))
(COND ((AND AsternGuarding
            AsternThrottle
            (<  HPTurbineExhaustTemp TurbineHeaderTemp))
       (SETQ HPTurbineExhaustTemp TurbineHeaderTemp))
      ((< HPTurbineExhaustTemp 0.0)
```

```
        (SETQ HPTurbineExhaustTemp 0.0)))

;; Next portion claims to be for the condenser.
(SETQ TurbineExhaustEnthalpy 1400.0)      ; move to Initialization?
(SETQ AsternSteamFlow (*$ SteamFlowRatioAstern .000278))
(SETQ c (*$ SteamFlowRatioForward .00046083 SteamFlowRatioAstern))
(COND ((= c 0.0) (SETQ LPTurbineExhaustTemp 0.0))
      (T (SETQ LPTurbineExhaustTemp
               (//$ (+$ (*$ LPTurbineExhaustTempCondenser
                            SteamFlowRatioForward)
                        (*$ .24885 SteamFlowRatioAstern))
                    c))))
(SETQ ShaftTorqueAstern
      (-$ (*$ SteamFlowRatioAstern ShaftRotation -.0002)
          (*$ SteamFlowRatioAstern .00065)))
(SETQ TotalShaftTorque
      (-$ (+$ ShaftTorqueForward ShaftTorqueAstern)
          (*$ ShaftRotation
              (+$ ShaftFrictionCoeff1
                  (//$ ShaftFrictionCoeff2
                       (+$ 1.0 (*$ (ABS ShaftRotation)
                                   ShaftFrictionCoeff3)))))))
(SETQ ShaftRotation
      (+$ ShaftRotation
          (//$ (*$ DeltaT 1.0125E-6
                   (-$ (*$ TotalShaftTorque 7506.0) PropTorque))
               (+$ 1.0 DeltaT))))
(COND ((< (ABS ShaftRotation) ShaftLockingSpeedWindow)
       (SETQ ShaftRotation 0.0)))
(SETQ ShaftRotationRPM (*$ ShaftRotation 60.0))
;; ETTNMI not here.
(SETQ ReverseShaftDirection (< ShaftRotation 0.0))
(SETQ ReductionGearNoiseAuralCue
      (AND CasualtyReductionGearNoise (NOT (= ShaftRotationRPM 0.0))))
(SETQ LPSentinelValveAuralCue
      (NOT (< (*$ LPTurbineExhaustPresCondenser .491) 20.0)))
(SETQ PropSpeedAdvance (*$ .95 ShipSpeed))
(SETQ temp (+$ (**$ PropSpeedAdvance)
               (*$ 225.0 (**$ ShaftRotation))))
(COND ((= temp 0.0)
       (SETQ AdvanceCoeffFunction 0.0))
      (T
       (SETQ AdvanceCoeffFunction
             (FBOUNDS -1.0
                      (//$ (*$ 15.0 ShaftRotation) (SQRT temp))
                      1.0))))
; tag 3055
(COND ((< PropSpeedAdvance 0.0)
       (COND ((> AdvanceCoeffFunction -.74)
              (SETQ PropThrustCoeff
                    (+$ 1.23 (*$ .86 AdvanceCoeffFunction))))
             ((> AdvanceCoeffFunction -.95)
              (SETQ PropThrustCoeff
                    (+$ .634 (*$ .86 AdvanceCoeffFunction))))
```

```
                (T
                 (SETQ PropThrustCoeff
                       (+$ 3.86 (*$ 4.25 AdvanceCoeffFunction))))))
         ((< AdvanceCoeffFunction .5)
          (SETQ PropThrustCoeff
                (-$ (*$ .1 AdvanceCoeffFunction) .29)))
         ((NOT (< AdvanceCoeffFunction .8))
          (SETQ PropThrustCoeff
                (-$ (*$ 2.2 AdvanceCoeffFunction) 1.66)))
         (T
          (SETQ PropThrustCoeff
                (-$ (*$ 1.133 AdvanceCoeffFunction) .807))))
; tag 3056
(COND ((< PropSpeedAdvance 0.0)
       (COND ((< AdvanceCoeffFunction -.4)
              (COND ((NOT (< AdvanceCoeffFunction -.84))
                     (SETQ PropTorqueCoeff
                           (+$ (*$ .077 AdvanceCoeffFunction) .054)))
                    (T
                     (SETQ PropTorqueCoeff
                           (+$ (*$ .406 AdvanceCoeffFunction) .3313)))))
             (T
              (SETQ PropTorqueCoeff
                    (+$ (*$ .042 AdvanceCoeffFunction) .04)))))
         ((< AdvanceCoeffFunction .4)
          (SETQ PropTorqueCoeff
                (-$ (*$ .0393 AdvanceCoeffFunction) .036)))
         ((NOT (> AdvanceCoeffFunction .77))
          (SETQ PropTorqueCoeff
                (-$ (*$ .103 AdvanceCoeffFunction) .061)))
         (T
          (SETQ PropTorqueCoeff
                (-$ (*$ .2739 AdvanceCoeffFunction) .1939))))
; tag 3057
(SETQ PropOpenWaterThrust (*$ PropThrustCoeff temp 447.75))
(SETQ PropNetThrust (*$ .91 PropOpenWaterThrust))
(SETQ PropTorque (*$ PropTorqueCoeff temp 6716.25))
(COND ((> ShipSpeed 33.78)
       (COND ((> ShipSpeed 42.23)
              (SETQ ShipResistance
                    (-$ (*$ ShipSpeed 8866.26) 265266.6)))
             (T
              (SETQ ShipResistance
                    (-$ (*$ ShipSpeed 4142.0) 64917.0)))))
         ((> ShipSpeed 13.51)
          (SETQ ShipResistance (-$ (*$ ShipSpeed 3305.37) 36655.6)))
         ((< ShipSpeed 0.0)
          (SETQ ShipResistance (*$ ShipSpeed 3578.45)))
         (T
          (SETQ ShipResistance (*$ ShipSpeed 592.0))))
; tag 3059
(SETQ ShipAdvanceCoeff
      (*$ 3.3266E-6 (-$ PropNetThrust ShipResistance)))
(SETQ ShipSpeed
```

```
        (+$ ShipSpeed (*$ ShipAdvanceCoeff DeltaT)))
    (SETQ ShipSpeedKnots
        (*$ .592 ShipSpeed))
    (COND ((EQ AheadEnable 'SECURED)
        (COND ((OR (> MSValveStrokeForward .143)
                   (AND SteamChestWarmupBypass1
                        SteamChestWarmupBypass2))
               (SETQ AheadSteamChestPresRTI
                  (//$ (*$ AheadSteamChestPresTC AheadSteamChestPres)
                       (+$ AheadSteamChestPresTC DeltaT))))))
        ((EQ AheadEnable 'OPERATING)
        (SETQ AheadSteamChestPresRTI AheadSteamChestPres))
        ((> MSValveStrokeForward .143)              ; assuming WARMUP
        (SETQ AheadSteamChestPresRTI
            (//$ (*$ AheadSteamChestPres DeltaT)
                 (+$ WarmupTC DeltaT))))
        (T
        (SETQ AheadSteamChestPresRTI
            (//$ (+$ (*$ AheadSteamChestRTI WarmupTC)
                     (*$ AheadSteamChestPres DeltaT))
                 (+$ WarmupTC DeltaT)))))))


(defun MainTurbineInit ()
    (setq AsternGuarding (select-ic '(nil nil t t)))    ; also in MainSteam
    (setq AheadGuarding (select-ic '(nil nil t t)))
    ; AheadGuarding is also in MainSteam and Controlair
    (setq MSValveStroke (select-ic '(0.0 0.0 0.0 0.0)))
    (setq MSValveStrokeForward
        (select-ic '(0.0 0.0 0.276 0.57)))    ; also in MainSteam
    (SETQ AheadSteamChestRTI 0.0)
    (SETQ MainShaftNormalizedRPM 0.0)
    (SETQ CasualtyMainThrottleInoperative NIL)
    (SETQ ForwardThrottle NIL)
    (SETQ AsternThrottle NIL)
    (SETQ MainThrottleForwardValveLift 0.0)
    (SETQ StrokeCrossbarMainThrottleForward 0.0)
    (SETQ AheadEnable 'SECURED)
    (SETQ AsternEnable 'SECURED)
    (SETQ SteamFlowRatioForward 0.0)
    (SETQ FB 0.0)
    (SETQ FD 0.0)
    (SETQ SupplySteamTempFunction 0.0)
    (SETQ SupplySteamPresFunction 0.0)
    (SETQ SteamFlowModified 0.0)
    (SETQ TurbineDeltaPresExit 0.0)
    (SETQ TurbineDeltaEnthalpyExit 0.0)
    (SETQ LPTurbineExhaustEnthalpy 0.0)
    (SETQ ForwardSteamFlow 0.0)
    (SETQ AsternSteamFlow 0.0)
    (SETQ HPTurbineExhaustPres 0.0)
    (SETQ Stage1HPPres 0.0)
    (SETQ Stage1HPTemp 0.0)
    (SETQ HPTurbineExhaustTemp 0.0)
```

```
    (SETQ ShaftTorqueForward 0.0)
    (SETQ MainThrottleAsternValveLift 0.0)
    (SETQ SteamFlowRatioAstern 0.0)
    (SETQ ShaftTorqueAstern 0.0)
    (SETQ TotalShaftTorque 0.0)
    (SETQ ShaftRotationRPM 0.0)
    (SETQ ReverseShaftDirection NIL)
    (SETQ AdvanceCoeffFunction 0.0)
    (SETQ PropThrustCoeff 0.0)
    (SETQ PropTorqueCoeff 0.0)
    (SETQ PropOpenWaterThrust 0.0)
    (SETQ PropNetThrust 0.0)
    (SETQ ShipSpeedKnots 0.0)
    (SETQ ShipAdvanceCoeff 0.0)
    (SETQ ShipResistance 0.0)
    (SETQ CasualtyReductionGearNoise NIL)
    (SETQ LPTurbineExhaustPresCondenser 5.0)
    (SETQ ShipSpeed 0.0)
    (SETQ PropTorque 0.0)
    (SETQ ShaftRotation 0.0)
    (SETQ TurbineExitEnthalpy 0.0)
    (SETQ LPTurbineExhaustPres 0.0)
    (SETQ LPTurbineExhaustTemp 70.0)
    (SETQ AheadSteamChestPresTC 20.0)
    (SETQ MinValveSupplySteamPres 400.0)
    (SETQ MinValveSupplySteamTemp 600.0)
    (SETQ SupplySteamTransientTemp .0006)
    (SETQ SteamFlowConstantReverse 2.5)
    (SETQ ShaftFrictionCoeff1 .1)
    (SETQ ShaftFrictionCoeff2 20.0)
    (SETQ ShaftFrictionCoeff3 10.0)
    (SETQ ShaftLockingSpeedWindow .1)
    (SETQ WarmupTC 5.0)
    ())

(version)
```

B.16 MCCircPump

```
(defun CircPumpRun ()
    (COND ((OR (NOT CircPumpPowerOn)
               CasualtyCircPumpMotorFail CircPumpStop
               (NOT (XOR CircPumpHighStart CircPumpLowStart)))
           ; tag 3709
           (SETQ CircPumpMotorSpeed 0.0)
           (SETQ AuralCueCircPumpOn//Off NIL)
           (SETQ CircPumpHighSpeedIndicator NIL)
           (SETQ CircPumpLowSpeedIndicator NIL))
          ((AND CircPumpHighStart (NOT CircPumpLowStart))
           (SETQ CircPumpMotorSpeed 880.0)
```

```
            (SETQ AuralCueCircPumpHi//Lo T)
            (SETQ AuralCueCircPumpOn//Off T)
            (SETQ CircPumpHighSpeedIndicator T)
            (SETQ CircPumpLowSpeedIndicator NIL))
           ((AND CircPumpLowStart (NOT CircPumpHighStart))
            (SETQ CircPumpMotorSpeed 440.0)
            (SETQ AuralCueCircPumpHi//Lo NIL)
            (SETQ AuralCueCircPumpOn//Off T)
            (SETQ CircPumpLowSpeedIndicator T)
            (SETQ CircPumpHighSpeedIndicator NIL)))
    ())


(defun CircPumpInit ()
    (setq CircPumpLowStart (select-ic '(nil nil nil nil)))
    (setq CircPumpHighStart (select-ic '(nil nil nil nil)))
    (setq CircPumpStop (select-ic '(nil nil nil nil)))

    (SETQ CasualtyCircPumpMotorFail NIL)
    (SETQ AuralCueCircPumpOn//Off NIL)
    (SETQ CircPumpHighSpeedIndicator NIL)
    (SETQ CircPumpLowSpeedIndicator NIL)
    (SETQ CircPumpMotorSpeed 0.0)
    ())

(defun TurnOnCircPump ()
    (SETQ CircPumpHighStart T))

(defun CircPumpStatus ()
    (statusprint CircPumpStatusVars))

(setq CircPumpStatusVars
      '(CircpumpMotorSpeed))

(version)
```

B.17 MCCondPumps

```
(defun MCCondPumpControlRun ()
    (COND ((OR (NOT CondPumplAPowerOn)
               CasualtyCondPumplA
               (AND (NOT StartCondPumplA)
                    (NOT StopCondPumplA)
                    LatchRelayFlag))
           (SETQ CondPumplAAuralCue NIL))
          (StopCondPumplA
           (COND ((NULL LatchRelayFlag) (SETQ LatchRelayFlag T)
                                        (SETQ CondPumplAAuralCue NIL))))
          (StartCondPumplA
           (COND (LatchRelayFlag (SETQ LatchRelayFlag NIL)
                                 (SETQ CondPumplAAuralCue T))))
```

```
        (T
         (SETQ CondPump1AAuralCue T)))
    (COND ((OR (NOT CondPump1BPowerOn)
               CasualtyCondPump1B
               (AND (NOT StartCondPump1B)
                    (NOT StopCondPump1B)
                    LatchRelayFlag2))
           (SETQ CondPump1BAuralCue NIL))
          (StopCondPump1B
           (COND ((NULL LatchRelayFlag2) (SETQ LatchRelayFlag2 T)
                                         (SETQ CondPump1BAuralCue NIL))))
          (StartCondPump1B
           (COND (LatchRelayFlag2 (SETQ LatchRelayFlag2 NIL)
                                  (SETQ CondPump1BAuralCue T))))
          (T
           (SETQ CondPump1BAuralCue T))))


(defun MCCondPumpControlInit ()
    (setq StopCondPump1B (select-ic '(nil nil nil nil)))
    (setq StopCondPump1A (select-ic '(nil nil nil nil)))
    (setq StartCondPump1B (select-ic '(nil nil nil nil)))
    (setq StartCondPump1A (select-ic '(nil nil nil nil)))

    (SETQ CondPump1AAuralCue NIL)
    (SETQ CondPump1BAuralCue NIL)
    (SETQ LatchRelayFlag NIL)
    (SETQ LatchRelayFlag2 NIL)
    (SETQ CasualtyCondPump1A NIL)
    (SETQ CasualtyCondPump1B NIL)
    (SETQ CondensatePumpGlandSealLoss NIL))


(declare (unspecial temp) (flonum temp))

(defun MCCondPumpsRun ()
  (PROG (temp)
    (MCCondPumpControlRun)
    ; Compute cavitation constant.
    (SETQ CondPumpNetPosSuctionHead (*$ .03588 HotwellWaterLevel))
    ; Calculate pump flow limit.
    (SETQ CavitationPumpFlowLimit
          (FBOUNDS 0.0
                   (-$ (*$ CavitationCurveSlope CondPumpNetPosSuctionHead)
                       CavitationCurveIntercept)
                   900.0))
    (SETQ CondensatePumpGlandSealLoss NIL)
    (SETQ CondensateRecircPathClosed NIL)
    (SETQ RecircFlowDemand 0.0)
    (SETQ Pump1AFlow 0.0)
    (SETQ Pump1BFlow 0.0)
    (SETQ RecircFlow 0.0)
    (SETQ CondensateFlow 0.0)
    (SETQ CondensateFlowAECondenser 0.0)
```

```
(SETQ PumpDischargePres 0.0)
(SETQ PumpDischargePres2 0.0)
(COND ((< HotwellWaterLevel .1)
  (SETQ RecircFlow 0.0)
  (SETQ CondensateFlow 0.0)
  (SETQ CondensateFlowAECondenser 0.0)
  (SETQ PumplADischargePres 0.0)
  (SETQ PumplBDischargePres 0.0))
 (T                                  ; This is the DO loop contents.
  (COND ((XOR CondPumplAAuralCue PumplAVentValve)
         (Malop 4 bit0)))
  (COND (PumplASuctionValve
    (COND ((NULL PumplAGlandSealValve)
           (SETQ CondensatePumpGlandSealLoss T)))
    ; test pump power on
    (COND ((NOT CondPumplAAuralCue)
           (SETQ PumplADischargePres HotwellBottomPresHead))
          (T
           (COND ((OR (> LPTurbineExhaustPresCondenser EPK7)
                      PumplAVentValve)
                  (COND ((> LPTurbineExhaustPresCondenser EPK7)
                         (SETQ EPC (*$ EPK6 HotwellWaterLevel))))
                  ; tag 3612
                  (SETQ PumplADischargePres
                        (+$ (*$ (-$ NoFlowPumpPres CoolingWaterPresHead)
                                (-$ 1.0 PumplADischargeValve))
                            CoolingWaterPresHead))
                  (SETQ TotalCondPumpHead
                        (FBOUNDS 0.0
                                 (-$ PumplADischargePres
                                     HotwellBottomPresHead)
                                 NoFlowPumpPres)))
                 (T
                  (SETQ CavitationPumpFlowLimit 0.0)
                  (SETQ TotalCondPumpHead 0.0)))
           ; tag 3613
           (COND (ValveAlignedForRecirc
                  (SETQ RecircFlowDemand
                        (*$ (+$ ThermostaticValveCoeff
                                ManualBypassValveCoeff)
                            (-$ CoolingWaterPresHead
                                HotwellBottomPresHead)
                            MaximumPumpFlow))
                  (COND ((< RecircFlowDemand 0.0)
                         (SETQ RecircFlowDemand 0.0)))))
           ; tag 3609
           (COND ((NOR CondensateSystemShutoffValve
                       ValveAlignedForRecirc)
                  (SETQ PumplADischargePres
                        (+$ NoFlowPumpPres HotwellBottomPresHead)))
                 (T
                  (COND (CondensateSystemShutoffValve
                         ;Compute single pump flow characteristic curve
                         (SETQ temp
```

```
                              (*$ 107.0 (SQRT (-$ NoFlowPumpPres
                                                  TotalCondPumpHead))))
                         (SETQ PumpFlowLoadIntercept
                               (*$ temp PumplADischargeValve)))
                        (ValveAlignedForRecirc      ; tag 3605
                         (SETQ PumpFlowLoadIntercept
                               (*$ 7.48
                                   MaxRecircFlowSinglePump
                                   PumplADischargeValve
                                   (+$ ThermostaticValveCoeff
                                       ManualBypassValveCoeff)))
                         (SETQ CondensateRecircPathClosed T)))
                  ; tag 3606
                  (COND ((> PumpFlowLoadIntercept
                            CavitationPumpFlowLimit)
                         (SETQ PumplAFlow CavitationPumpFlowLimit))
                        (T
                         (SETQ PumplAFlow PumpFlowLoadIntercept)))
                  (COND (CondensateRecircPathClosed
                         (SETQ PumplADischargePres
                               (-$ NoFlowPumpPres
                                   (*$ RecircPresConstant
                                       PumplAFlow))))))))))))
; Repeat for PumplB
(COND ((XOR CondPumplBAuralCue PumplBVentValve)
       (Malop 4 bit0)))
(COND (PumplBSuctionValve
  (COND ((NULL PumplBGlandSealValve)
         (SETQ CondensatePumpGlandSealLoss T)))
  ; test pump power on
  (COND ((NOT CondPumplBAuralCue)
         (SETQ PumplBDischargePres HotwellBottomPresHead))
        (T
         (COND ((OR (> LPTurbineExhaustPresCondenser EPK7)
                    PumplBVentValve)
                (COND ((> LPTurbineExhaustPresCondenser EPK7)
                       (SETQ EPC (*$ EPK6 HotwellWaterLevel))))
                ; tag 3612
                (SETQ PumplBDischargePres
                      (+$ (*$ (-$ NoFlowPumpPres CoolingWaterPresHead)
                              (-$ 1.0 PumplBDischargeValve))
                          CoolingWaterPresHead))
                (SETQ TotalCondPumpHead
                      (FBOUNDS 0.0 (-$ PumplBDischargePres
                                       HotwellBottomPresHead)
                               NoFlowPumpPres)))
               (T
                (SETQ CavitationPumpFlowLimit 0.0)
                (SETQ TotalCondPumpHead 0.0)))
         ; tag 3613
         (COND (ValveAlignedForRecirc
                (SETQ RecircFlowDemand
                      (*$ (+$ ThermostaticValveCoeff
                              ManualBypassValveCoeff)
```

```
                                (-$ CoolingWaterPresHead
                                    HotwellBottomPresHead)
                                MaximumPumpFlow))
                        (COND ((< RecircFlowDemand 0.0)
                               (SETQ RecircFlowDemand 0.0)))))
                ; tag 3609
                (COND ((NOR CondensateSystemShutoffValve
                            ValveAlignedForRecirc)
                       (SETQ Pump1BDischargePres
                             (+$ NoFlowPumpPres HotwellBottomPresHead)))
                      (T
                       (COND (CondensateSystemShutoffValve
                              ;Compute single pump flow characteristic curve
                              (SETQ temp
                                    (*$ 107.0 (SQRT (-$ NoFlowPumpPres
                                                        TotalCondPumpHead))))
                              (SETQ PumpFlowLoadIntercept
                                    (*$ temp Pump1BDischargeValve)))
                             (ValveAlignedForRecirc
                              ; tag 3605
                              (SETQ PumpFlowLoadIntercept
                                    (*$ 7.48
                                        MaxRecircFlowSinglePump
                                        Pump1BDischargeValve
                                        (+$ ThermostaticValveCoeff
                                            ManualBypassValveCoeff)))
                              (SETQ CondensateRecircPathClosed T)))
                       ; tag 3606
                       (COND ((> PumpFlowLoadIntercept
                                 CavitationPumpFlowLimit)
                              (SETQ Pump1BFlow CavitationPumpFlowLimit))
                             (T
                              (SETQ Pump1BFlow PumpFlowLoadIntercept)))
                       (COND (CondensateRecircPathClosed
                              (SETQ Pump1BDischargePres
                                    (-$ NoFlowPumpPres
                                        (*$ RecircPresConstant
                                            Pump1BFlow)))))))))))
        ; after the DO loop
        (SETQ CondensateFlow (+$ Pump1AFlow Pump1BFlow))
        (COND (CondensateRecircPathClosed
               (SETQ CondensateFlowAECondenser (*$ CondensateFlow .13369)))
              (T
               (SETQ CondensateFlowAECondenser
                     (MIN RecircFlowDemand (*$ CondensateFlow .13369)))))
        (SETQ RecircFlow CondensateFlowAECondenser))))


(defun MCCondPumpsInit ()

    (MCCondPumpControlInit)

    (setq CondensateSystemShutoffValve (select-ic '(nil nil t t)))
    (setq Pump1BVentValve (select-ic '(nil nil nil t)))
```

```
    (setq PumplAVentValve (select-ic '(nil nil t t)))
    (setq PumplBGlandSealValve (select-ic '(nil nil nil t)))
    (setq PumplAGlandSealValve (select-ic '(nil nil t t)))
    (setq PumplBSuctionValve (select-ic '(nil nil nil t)))
    (setq PumplASuctionValve (select-ic '(nil nil t t)))
    (setq PumpDischargeValve2 (select-ic '(0.0 0.0 1.0 1.0)))
    (setq PumpDischargeValve1 (select-ic '(0.0 0.0 1.0 1.0)))

    (SETQ CoolingWaterPresHead 0.0)      ; This may be wrong.
    (SETQ CondPumpNetPosSuctionHead 0.0)
    (SETQ CondensateFlow 0.0)
    (SETQ CondensateRecircPathClosed NIL)
    (SETQ CondensateFlowAECondenser 0.0)
    (SETQ RecircFlow 0.0)
    (SETQ CavitationPumpFlowLimit 0.0)
    (SETQ EPK6 15.0)
    (SETQ CavitationCurveSlope 1395.0)
    (SETQ CavitationCurveIntercept 300.0)
    (SETQ NoFlowPumpPres 76.0)
    (SETQ MaximumPumpFlow 42.0)
    (SETQ MaxRecircFlowSinglePump 18.0)
    (SETQ RecircPresConstant .2)
    (SETQ EPK7 26.0)
    ())

(defun TurnOffCondensatePumps ()
    (SETQ StartCondPumplA NIL)
    (SETQ StartCondPumplB NIL)
    (SETQ StopCondPumplA T)
    (SETQ StopCondPumplB T))

(defun TurnOnCondensatePumps ()
    (SETQ StartCondPumplA T)
    (SETQ StartCondPumplB T)
    (SETQ StopCondPumplA NIL)
    (SETQ StopCondPumplB NIL))

(version)
```

B.18 MCRecirc

```
(declare (unspecial temp) (flonum temp))

(defun MCRecircRun ()
  (PROG (temp)
    ; tag 3294
    ; First Clause to come depends on
    ; CommonCondenserFill, which in independent mode is
    ; always T.
    (COND ((NOT MCFillStopValve)
           (SETQ CondenserFillFlow 0.0))
```

```
        ((OR CondenserRecircValveHi CondenserRecircValveLo)
         (SETQ temp (-$ FeedwaterCoolerPres
                        (+$ HotwellBottomPresHead 15.0)))
         (COND ((< temp 0.0)
                (SETQ CondenserFillFlow FillFlowRateConstant))
               (T
                (SETQ CondenserFillFlow (*$ FillFlowRateConstant
                                            (SQRT temp))))))
        (T
         (SETQ CondenserFillFlow FillFlowRateConstant)))
    ; tag 3295
    (COND ((OR CondenserRecircValveHi CondenserRecircValveLo)
           (SETQ ValveAlignedForRecirc T)
           (COND ((AND CondenserRecircValveHi CondenserRecircValveLo)
                  (Malop 4 bitl))))
          (T
           (SETQ ValveAlignedForRecirc NIL)))
    ; tag 3296
    (SETQ ManualBypassValveCoeff (*$ ManualFlowBypassValveCoeff
                                     ManualRecircBypassValve))
    (COND ((> CoolingWaterTemp 150.0)
           (COND (CasualtyRecircValveFailClosed
                  (SETQ ThermostaticValveCoeff 0.0))
                 (T
                  (SETQ ThermostaticValveCoeff 1.0))))
          ((> RecircValveOpenTemp CoolingWaterTemp)
           (SETQ ThermostaticValveCoeff 0.0))
          (T                              ; Instead of LE, tag 3291
           (COND (CasualtyRecircValveFailClosed
                  (SETQ ThermostaticValveCoeff 0.0))
                 (T
                  (SETQ ThermostaticValveCoeff
                        (FBOUNDS 0.0
                                 (//$ (-$ CoolingWaterTemp
                                          RecircValveOpenTemp)
                                      (-$ 150.0 RecircValveOpenTemp))
                                 1.0)))))))


(defun MCRecircInit ()
    (setq ManualRecircBypassValve (select-ic '(0.0 0.0 0.0 0.0)))
    (setq MCFillStopValve (select-ic '(nil nil nil nil)))
    (setq CondenserRecircValveLo (select-ic '(nil nil t t)))
    (setq CondenserRecircValveHi (select-ic '(nil nil nil nil)))
    (SETQ CasualtyRecircValveFailClosed NIL)

    (SETQ CommonCondenserFill T)        ; assuming independent mode
    (SETQ FeedwaterCoolerPres 30.0)     ; assuming independent mode
    (SETQ FillFlowRateConstant 5.0)
    (SETQ ManualFlowBypassValveCoeff 1.0)
    (SETQ RecircValveOpenTemp 140.0)
    ())

(version)
```

B.19 MCSeaCirc

```
(declare (unspecial a b temp) (flonum a b temp))

(defun MCSeaCircRun ()
 (PROG (a b temp)
   (COND ((< ShipSpeedKnots 0.0) (SETQ temp 0.0))
         (T                      (SETQ temp ShipSpeedKnots)))
   (store (TSPres 1) (*$ PumpFlowCoeff (**$ CircPumpMotorSpeed)))
   (store (TSPres 2) (*$ ScoopFlowCoeff (**$ temp)))
   (store (TSFlowRes 1) (+$ (*$ ResFunSlopel CircPumpMotorSpeed)
                            ResFunIntercept))
   (store (TSFlowRes 2) (+$ (*$ ResFunSlope temp)
                            ResFunIntercept0))
   (store (TSFlowRes 3) (//$ (*$ (TSFlowRes 1) (TSFlowRes 2))
                             (+$ (TSFlowRes 1) (TSFlowRes 2))))
   (store (TSPres 3) (//$ (+$ (*$ (TSPres 1) (TSFlowRes 2))
                              (*$ (TSPres 2) (TSFlowRes 1)))
                          (+$ (TSFlowRes 1) (TSFlowRes 2))))
   (SETQ temp (*$ MCSWValveCoeff DischargeButterflyValvePos))
   (SETQ temp (//$ temp (+$ 1.0 (*$ temp MCSWValveFactor))))
   (SETQ a (SQRT temp))
   ; at DO loop
   (for i from 1 to 3 do
      (SETQ b (*$ a .5 (+$ MCSWResConstant (TSFlowRes i))))
      (store (TCFlow i) (*$ (-$ (SQRT (+$ (**$ b) (TSPres i))) b) a))
      (store (TCPres i) (-$ (TSPres i) (*$ (TSFlowRes i) (TCFlow i)))))
   (COND ((AND PumpButterflyValve PumpFlapperOpen)
          (COND ((AND ScoopButterflyValve ScoopFlapperOpen)
                 (SETQ MCSWCondFlow (TCFlow 3))
                 (SETQ MCSWCondPres (TCPres 3)))
                (T
                 (SETQ MCSWCondFlow (TCFlow 1))
                 (SETQ MCSWCondPres (TCPres 1)))))
         ((AND ScoopButterflyValve ScoopFlapperOpen)
          (SETQ MCSWCondFlow (TCFlow 2))
          (SETQ MCSWCondPres (TCPres 2)))
         (T
          (SETQ MCSWCondFlow 0.0)
          (SETQ MCSWCondPres 0.0)))
   (SETQ CondenserWaterBoxPres (+$ MCSWCondPres StaticCondenserPresReading))
   (SETQ SeaWaterCoolantMassFlow (*$ MCSWCondFlow .13369))
   ; start of branching to 3320
   (COND ((AND PumpButterflyValve PumpFlapperOpen)      ; tag 3310
          (SETQ temp (//$ (-$ (TSPres 1) MCSWCondPres) (TSFlowRes 1)))
          (SETQ SCPumpDischargePres (+$ (*$ MCSWPumpCoeff temp)
                                        MCSWCondPres
                                        StaticPumpSuctionReading))
          (SETQ SWPumpSuctionPres (-$ StaticPumpSuctionReading
                                      (*$ (-$ ResFunIntercept
                                              MCSWPumpCoeff)
                                          temp))))
         ((NOR PumpButterflyValve PumpFlapperOpen)      ; tag 3311
```

```
          (SETQ a (//$ 1.0 (+$ 1.0 MCSWComplianceConstant)))
          (SETQ temp (+$ SWPumpSuctionPres (*$ a SCPumpDischargePres)))
          (SETQ SWPumpSuctionPres
                (*$ (-$ temp (*$ MCSWComplianceConstant (TSPres 1))) a))
          (SETQ SCPumpDischargePres
                (*$ (+$ temp (TSPres 1)) a)))
         (PumpButterflyValve              ; tag 33.2
          (SETQ SWPumpSuctionPres StaticPumpSuctionReading)
          (SETQ SCPumpDischargePres
                (+$ StaticPumpDischargeReading (TSPres 1))))
         (T
          (SETQ SWPumpSuctionPres
                (-$ (+$ StaticPumpSuctionReading MCSWCondPres) (TSPres 1)))
          (SETQ SCPumpDischargePres
                (+$ StaticPumpDischargeReading MCSWCondPres))))
    ; tag 3320
    (SETQ a (-$ MCSWCondPres (TSPres 1)))
    (COND ((> a 0.0)
           (SETQ PumpFlapperPosCommand NIL)
           (COND ((< a PumpFlapperBangingWidth)
                  (SETQ PumpFlapperBanging T))
                 (T
                  (SETQ PumpFlapperBanging NIL))))
          (T
           (SETQ PumpFlapperPosCommand T)))
    ; tag 3330
    (SETQ a (-$ MCSWCondPres (TSPres 2)))
    (COND ((> a 0.0)
           (SETQ ScoopFlapperPosCommand NIL)
           (COND ((< a ScoopFlapperBangingWidth)
                  (SETQ ScoopFlapperBanging T))
                 (T
                  (SETQ ScoopFlapperBanging NIL))))
          (T
           (SETQ ScoopFlapperPosCommand T)))
    ; tag 3390
    ; here things get VERY confused!
    ; tag 3330
    ))


(defun MCSeaCircInit ()
    (setq DischargeButterflyValvePos (select-ic '(0.0 0.0 1.0 1.0)))
    (setq ScoopFlapperQuiteOpen (select-ic '(t t nil nil)))
    (setq PumpFlapperOpen (select-ic '(nil nil nil nil)))
    (setq ScoopFlapperOpen (select-ic '(nil nil t t)))
    (setq PumpButterflyValve (select-ic '(nil nil t t)))
    (setq ScoopButterflyValve (select-ic '(nil nil t t)))

    (SETQ PumpFlapperPosCommand NIL)
    (SETQ PumpFlapperPosEnable NIL)
    (SETQ PumpFlapperBanging NIL)
    (SETQ ScoopFlapperPosCommand NIL)
    (SETQ ScoopFlapperPosEnable NIL)
```

```
    (SETQ ScoopFlapperBanging NIL)

    (FILLARRAY TCPres '(0.0))
    (FILLARRAY TCFlow '(0.0))
    (FILLARRAY TSFlowRes '(0.0))
    (FILLARRAY TSPres '(0.0))

    ; model constants
    (SETQ MCSWComplianceConstant 2.0)
    (SETQ StaticPumpSuctionReading 3.1)
    (SETQ StaticPumpDischargeReading 3.1)
    (SETQ StaticCondenserPresReading 3.1)
    (SETQ MCSWPumpCoeff .0001667)
    (setq MCSWCondFlow 0.0)
    (SETQ ScoopFlapperBangingWidth 0.0)
    (SETQ PumpFlapperBangingWidth 0.0)
    (SETQ SecuredCondenserPres 0.0)
    (SETQ SecuredCondenserFlow 0.0)
    (SETQ MCSWResConstant 6.24E-5)
    (SETQ MCSWValveFactor 6.703E-9)
    (SETQ MCSWValveCoeff 1.0E10)
    (SETQ ScoopFlowCoeff .02169)
    (SETQ ResFunSlope 1.6275E-5)
    (SETQ ResFunIntercept0 4.72E-5)
    (SETQ PumpFlowCoeff 2.289E-5)
    (SETQ ResFunSlope1 9.592E-7)
    (SETQ ResFunIntercept .000225)
    (SETQ SWPumpSuctionPres 0.0)
    (SETQ SCPumpDischargePres 0.0)
    (SETQ CondenserWaterBoxPres 0.0)
    ())

(version)
```

B.20 MTGlandSeal

```
(declare (unspecial temp a b c d) (flonum temp a b c d))

(defun MTGlandSealRun ()
    (PROG (temp a b c d)
      (COND ((> LPTurbineExhaustPresCondenser 20.0)
             (SETQ a (-$ (*$ GlandSteamLeakoffCoeff MainShaftNormalizedRPM)
                         1.0)))
            (T
             (SETQ a (+$ (*$ GlandSteamLeakoffCoeff MainShaftNormalizedRPM)
                         (*$ .015 LPTurbineExhaustPresCondenser)
                         -.4))))

      ; leakoff done, now the bypass valve
      (COND ((AND 150lbSteamEnable (> SupplyBypassValve .1))
             (SETQ b (*$ BypassValveCoeff (-$  SupplyBypassValve .1))))
```

```
       (T
        (SETQ b 0.0)))              ; Override
(COND (JackHandleOperative
       (SETQ c (*$ UnloadingValveMOCoeff
                   (-$ UnloadingValveStrokeIndicator
                       StemPositionUnloadingValve))))
       (T
        (SETQ c 0.0)))
(SETQ d (+$ a b))      ; start with valve auto enable flags off
(SETQ EYKUV NIL)
(SETQ EYKSV NIL)
(SETQ EYP50C 0.0)                   ; These are locals, I suppose
(COND ((AND SenseSupplyValve (= SupplyValveCAPres 20.0))
       ; tag 4251 - supply enabled
       (SETQ EYKSV T))
       (T                           ; supply disabled - failed open
        (SETQ EYXSV T)))
(COND ((AND CasualtyGlandSteamLoss (= UnloadingValveCAPres 20.0))
       (COND (SenseUnloadingValve (SETQ EYKUV T))
             (T                    (SETQ EYXUV NIL))))
       (T
        (SETQ EYXUV T)))
(COND (150lbSteamEnable
       (COND ((AND CasualtyGlandSteamLoss
                   (= UnloadingValveCAPres 20.0))
              (COND (SenseUnloadingValve
                     (SETQ EYP50C (+$ d c 2.0)))
                    ((AND SenseSupplyValve
                          (= SupplyValveCAPres 20.0))
                     (SETQ EYP50C (+$ d 2.0)))
                    (T
                     (SETQ EYP50C 10.0))))
             (T
              (SETQ EYP50C (+$ d c -1.0)))))
       (T
        (SETQ EYP50C 0.0)))
; tag 4254
(SETQ EYP50C (MIN EYP50C 10.0))    ; unloading valve indication
(COND (EYKUV
       (COND ((> GlandManifoldSteamPres 3.0)
              (SETQ UnloadingValveStrokeIndicator 1.0))
             ((> GlandManifoldSteamPres 2.0)
              (SETQ UnloadingValveStrokeIndicator
                    (-$ GlandManifoldSteamPres 2.0)))
             (T
              (SETQ UnloadingValveIndicator 0.0)))))
; tag 4255 - supply valve indicator
(COND (EYKSV
       (COND ((> GlandManifoldSteamPres 2.0)
              (SETQ SupplyValveStrokeIndicator 0.0))
             ((> GlandManifoldSteamPres 1.0)
              (SETQ SupplyValveStrokeIndicator
                    (-$ 2.0 GlandManifoldSteamPres)))
             (T
```

```
                    (SETQ SupplyValveStrokeIndicator 1.0)))))
      ; tag 4256 - updating pressure gages
      (COND ((= UnloadingValveCAPres 0.0)
             (SETQ UnloadingValveActuationAirPres 0.0)
             (SETQ UnloadingValveAirPres NIL))
            (T
             (SETQ UnloadingValveActuationAirPres
                   (-$ ValveCAPresMax (*$ 7.0 UnloadingValveStrokeIndicator)))
             (SETQ UnloadingValveAirPres T)))
      (COND ((= SupplyValveCAPres 0.0)
             (SETQ SupplyValveActuationAirPres 0.0)
             (SETQ SupplyValveAirPres NIL))
            (T
             (SETQ SupplyValveActuationAirPres
                   (+$ (*$ 7.0 (-$ 1.0 SupplyValveStrokeIndicator))
                       ValveCAPresMin))
             (SETQ SupplyValveAirPres T)))
      (SETQ GlandManifoldSteamPres
            (//$ (+$ (*$ GlandManifoldSteamPres GlandSteamPresTC)
                     (*$ EYP50C DeltaT))
                 (+$ GlandSteamPresTC DeltaT)))
      (COND ((< GlandManifoldSteamPres .1)
             (SETQ CasualtyGlandSealLoss T)
             (SETQ GlandSteamFlow 0.0)
             (SETQ GlandSteamEnthalpy 0.0)
             (SETQ GlandManifoldSteamPres 0.0))
            (T
             (SETQ CasualtyGlandSealLoss NIL)
             (SETQ GlandSteamFlow .12)
             (SETQ GlandSteamEnthalpy 1200.0)))
      (COND ((OR (> GlandManifoldSteamPres 9.0)
                 (NULL SenseUnloadingValve2))
             (Malop 1 Bit7)))))


(defun MTGlandSealInit ()
   (setq SupplyBypassValve (select-ic '(0.0 0.0 0.0 0.0)))
   (setq StemPositionUnloadingValve (select-ic '(1.0 1.0 1.0 1.0)))
   (setq SenseUnloadingValve2 (select-ic '(nil nil t t)))
   (setq SenseUnloadingValve (select-ic '(nil nil t t)))
   (setq SenseSupplyValve (select-ic '(nil nil t t)))
   (setq JackHandleOperative (select-ic '(nil nil nil nil)))
   (SETQ SupplyValveAirPres NIL)
   (SETQ UnloadingValveAirPres NIL)
   (SETQ SupplyValveCAPres 0.0)

   ;; Model Constants
   (SETQ GlandManifoldSteamPres 0.0)
   (SETQ ValveCAPresMin 13.0)
   (SETQ ValveCAPresMax 20.0)
   (SETQ BypassValveCoeff 3.0)
   (SETQ UnloadingValveMOCoeff 4.0)
   (SETQ GlandSteamPresTC 15.0)
   (SETQ GlandSteamLeakoffCoeff .84)
```

```
    ())

;(defun TurnOnMTGlandSeal ()
;     (SETQ SupplyValveStrokeIndicator .5)
;     (SETQ UnloadingValveStrokeIndicator .5))

(version)
```

END

FILMED

4-86

DTIC